

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Управление корпоративными информационными процессами
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Анализ механизмов управления платформой сообщества для поддержки интегрированного проектирования обучения»

Обучающийся

П. И. Баринов
(Инициалы Фамилия)

(личная подпись)

Научный
руководитель

канд. пед. наук, доцент, Т. А. Агошкова
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Содержание

Введение.....	4
1 Анализ существующих работ по теме механизмов управления платформой сообщества преподавателей.....	7
1.1 CSCL, ориентированная на знания.....	7
1.1.1 Моделирование взаимодействия с пользователем.....	9
1.1.2 Управление данными о взаимодействии.....	12
1.2 Инженерное программное обеспечение CSCL-приложения.....	13
1.2.1 Системы группового обучения.....	14
1.2.2 Методы разработки программного обеспечения.....	17
1.3 Распределенные и grid-вычисления для удовлетворения современных потребностей в CSCL.....	25
2 Проведение анализа существующих решений в рамках рассматриваемой тематики.....	28
2.1 Проведение анализа работы существующих решений различных площадок.....	28
2.2 Результаты сравнительного анализа существующих решений.....	33
3 Вычислительная модель CSCL на основе повторного использования.....	38
3.1 Универсальное программирование.....	38
3.1.1 Цели универсального программирования.....	39
3.1.2 Пример приложения CSCL с использованием методик универсального программирования.....	41
3.2 Общий взгляд на приложения для совместного обучения.....	47
3.2.1 Общая основа для приложений CSCL.....	47
3.2.2 Важность анализа взаимодействия в практике CSCL.....	50
3.3 Вычислительная модель для анализа данных и управления ими в Библиотеке совместного обучения.....	53
3.3.1 Архитектура Библиотеки совместного обучения.....	55
3.3.2 О преимуществах использования сервис-ориентированных	

архитектур для CSCL.....	61
3.4 Веб-сервисы и grid-технологии для улучшения CSCL	64
3.4.1 Общие grid-сервисы.....	67
3.4.2 Прикладные grid-сервисы	68
3.4.3 Основные сервисы сетевой инфраструктуры	69
4 Объединение CSCL, универсального программирования и распределенных вычислений: оценка и результаты	71
4.1 Проведение эксперимента и результаты распараллеливания обработки файлов журнала.....	71
4.2 Результаты распараллеливания обработки файлов журнала с использованием JXTA	75
4.3 Анализ результатов распараллеливания обработки файлов журнала ...	78
4.4 Продвинутая разработка программного обеспечения с использованием Библиотеки совместного обучения	80
Заключение	83
Список используемой литературы	85

Введение

Совместное обучение с компьютерной поддержкой (CSCL) - одна из наиболее влиятельных исследовательских парадигм, направленных на улучшение преподавания и усвоения знаний с помощью современных информационно-коммуникационных технологий. Приложения CSCL направлены на создание виртуальной среды совместного обучения, в которой учащиеся и преподаватели могут сотрудничать друг с другом для достижения общей цели обучения.

Одной из ключевых проблем при разработке приложений CSCL является анализ данных взаимодействия, который является основной функцией для поддержки обучения и оценки в средах CSCL. Он опирается на информацию, полученную в результате действий, выполняемых участниками в ходе процесса совместной работы. Эффективное внедрение этой информации и извлеченных знаний в приложения CSCL создает основу для повышения осведомленности о мониторинге и обратной связи для достижения успешного процесса обучения в условиях совместной работы.

Современные среды CSCL больше не зависят от однородных групп, статичного контента и ресурсов, а также от единой педагогики, но в этом контексте необходима высокая индивидуализация и гибкость, а это означает, что методы совместного обучения необходимо постоянно адаптировать, корректировать и персонифицировать для каждой конкретной целевой учебной группы. Удовлетворение этих потребностей в области CSCL представляет собой серьезную проблему для исследовательского сообщества.

Объектом исследования является поддержка интегрированного проектирования обучения.

Предметом исследования является механизмы управления платформой сообщества преподавателей для поддержки интегрированного проектирования обучения.

Гипотеза исследования заключается в следующем: механизм

управления платформой сообщества преподавателей для поддержки интегрированного проектирования обучения будет универсальным, если будет смоделирована структура эффективного управления и анализа данных взаимодействия, генерируемых в ходе практики CSCL и предложен усовершенствованный метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL.

Целью исследования является моделирование структуры эффективного управления и анализа данных взаимодействия для создания универсального механизма разработки программного обеспечения на основе повторного использования компонентов приложений CSCL.

Для достижения поставленной цели и подтверждения гипотезы необходимо решить следующие задачи:

- проанализировать процесс эффективного сбора, обработки, анализа и представления информации и знаний, полученных в результате групповой деятельности, в приложениях CSCL;
- разработать концептуальную модель вычислительной системы в форме общей спецификации;
- смоделировать структуру эффективного управления и анализа данных взаимодействия, генерируемых в ходе практики CSCL;
- усовершенствовать метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL для полуавтоматического генерирования реализаций на основе независимых спецификаций;
- провести экспериментальную апробацию усовершенствованного механизма управления платформой сообщества преподавателей для поддержки интегрированного проектирования обучения на основе открытых источников.

На защиту магистерской диссертации выносятся следующие положения:

- структура эффективного управления и анализа данных взаимодействия, генерируемых в ходе практики CSCL;

- усовершенствованный метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL;
- результаты экспериментальной апробации механизма управления платформой сообщества для поддержки интегрированного проектирования обучения.

Научная новизна исследования заключается в том, что был предложен универсальный механизм управления платформой сообщества преподавателей для поддержки интегрированного проектирования обучения, который может быть адаптирован под любую спецификацию приложений CSCL.

Практическая значимость заключается в разработке усовершенствованного метода разработки программного обеспечения на основе повторного использования компонентов приложений CSCL для поддержки интегрированного проектирования обучения.

Методы исследования: сравнительный анализ существующих решений и моделирование.

Первый раздел посвящен определению CSCL, его проблем и путей их решения.

Во втором разделе производится анализ существующих решений в сфере CSCL.

Третий раздел посвящен разработке модели, для решения проблем, найденных в первых и вторых разделах.

В четвертом разделе исследования производится анализ и экспертиза валидности использования grid-систем.

Диссертация состоит из введения, четырех разделов, заключения и списка используемой литературы.

1 Анализ существующих работ по теме механизмов управления платформой сообщества преподавателей

1.1 CSCL, ориентированная на знания

При разработке компьютерных сред совместного обучения (CSCL), поддерживающих совместное обучение в режиме онлайн, необходимо учитывать несколько факторов, чтобы обеспечить полную поддержку онлайн-обучения. Одним из ключевых вопросов является управление взаимодействием и анализ данных для поддержки осведомленности, и оценки на основе информации, полученной в результате действий, выполняемых участниками в процессе совместной работы [3].

Успех приложений CSCL в значительной степени зависит от способности таких приложений внедрять информацию и знания, полученные в результате взаимодействия в групповой деятельности, и использовать их для достижения более эффективной оценки совместного обучения. Это стратегическое направление исследований все еще находится в зачаточном состоянии, поскольку вопросы, связанные с внедрением информации и знаний, недостаточно исследованы.

Первоначальный подход, который представлен в работе Zumbach J., et al., рассматривает использование обратной связи в онлайн-обучении и ее влияние на групповую деятельность в целом [32]. С этой целью некоторые типы информации, генерируемые в результате групповой деятельности, рассматриваются как релевантные знания, которые должны быть переданы членам группы в целях обратной связи, но процесс сбора информации, ее анализа и извлечения желаемых знаний не предусмотрен.

Автоматический сбор информации о взаимодействии представлен в работе Martinez A., et al. для поддержки процессов формирующей оценки и построения каркасов [24]. С этой целью они представляют гибкую, эффективную и универсальную модель сотрудничества, которая может

применяться в различных ситуациях CSCL. Затем они представляют эту модель с помощью вычислительной модели совместных действий на основе XML. Несмотря на то, что модель различает различные типы взаимодействий, исходящих из разных источников данных, не проводится никаких дальнейших исследований того, как классифицировать, обрабатывать и представлять эту информацию и полученные знания эффективным образом, чтобы обеспечить эффективную обратную связь с группами совместной работы, при одновременном создании самой модели.

В работе Daradoumis T., et al. [12] показан комплексный подход, включающий качественный, количественный анализ и анализ социальных сетей для анализа и оценки эффективности групп онлайн-обучения на основе информации, собранной в ходе долгосрочного совместного обучения. Этот подход определяет несколько важных высокоуровневых процессов совместного обучения, которые позволяют провести первоначальную классификацию и оценку взаимодействия в групповой деятельности с точки зрения учебного продукта, функционирования группы и социальной поддержки (т.е. каркасов):

- учебный продукт относится к навыкам выполнения заданий и знаниям, приобретенным каждым участником, а также к качеству групповой работы;
- групповое функционирование предполагает анализ и оценку как поведения членов группы при взаимодействии, так и социальных аспектов групповой работы;
- «каркасы» относятся к социальной поддержке среди участников, а также к помощи, ориентированной на выполнение задач или групповое функционирование, оказываемой участнику, который не совсем способен или готов выполнить задачу самостоятельно.

Однако представление высокого уровня, представленное в этом подходе, не очень помогает определить и определить точный смысл всех процессов обучения и низкоуровневых взаимодействий, происходящих в

повседневных виртуальных классах. Кроме того, все исследование было основано на среде базовой поддержки совместной работы, которая демонстрирует серьезную нехватку инструментов для сбора и анализа соответствующей информации, особенно в режиме реального времени, и, таким образом, оно было проведено после завершения задачи, что гораздо меньше влияет на групповую активность.

Недавно было проведено большое количество исследований с целью идентифицировать и моделировать точные процессы и показатели в соответствии с реальными навыками и намерениями, продемонстрированными участниками в контексте совместной деятельности [2], которые являются примером совместной учебной деятельности, используемой в данной диссертации и представляющей интерес для эмпирических целей. Далее представлен обзор текущих исследований о том, как моделировать дискуссионное взаимодействие, его последующую обработку и анализ.

1.1.1 Моделирование взаимодействия с пользователем

Schellens T. и Valcke M. исследовали, приводит ли совместное обучение в асинхронных дискуссионных группах к улучшению академического дискурса и формированию знаний [28]. Их исследовательская работа показала, что студенты в дискуссионных группах были ориентированы на выполнение задач и что наблюдалась более высокая доля высоких этапов формирования знаний. Более того, было обнаружено значительное увеличение когнитивного взаимодействия, ориентации на выполнение задач и более высоких фаз построения знаний.

Кроме того, важной проблемой, возникающей при совместном обучении, является переход от расхождений к общему пониманию и возможному накоплению знаний [5]. Суть в том, чтобы понять, как развиваются совместные взаимодействия с течением времени: чаще ли учащиеся поднимают новые вопросы (идеи) по мере того, как они лучше знакомятся с обсуждением и участниками дискуссии, и становится ли

накопление общих знаний богаче с течением времени, и последующие доказательства того, что учащиеся смогли сформировать свое собственное понимание на основе своего взаимодействия с другими учащимися. С этой целью модель комментирует и исследует различные элементы, которые способствуют пониманию природы совместных взаимодействий, такие как пассивность и активность учащихся, а также эффективность и влияние их вклада в достижение общей цели обсуждения.

Большие объемы информационных данных генерируются в результате асинхронного обсуждения, которое включает сложные вопросы совместной работы и процесса обучения (например, групповое благополучие), а также оценку собственной деятельности, деятельности других участников группы и групповой деятельности. С одной стороны, количественной информацией можно управлять, применяя структурированный процесс, в котором взаимодействия пользователей помечаются определенными показателями в соответствии с таксономией навыков совместной учебной беседы, которая моделирует различные типы взаимодействий на разных уровнях [1]. Более того, типичные количественные показатели эффективности и динамики участников (например, количество материалов, написанных и прочитанных каждым участником) также считаются релевантными для моделирования функционирования группы и выполнения задач.

Количественный контент-анализ все чаще используется для того, чтобы превзойти анализ на поверхностном уровне в CSCL (например, подсчет сообщений), и несколько схем контент-анализа были использованы для анализа стенограмм асинхронных онлайн-дискуссионных групп в формальных образовательных учреждениях [4]. Хотя этот метод исследования часто использовался, стандарты еще не установлены. Как следствие, эмпирическая база, подтверждающая действительность инструментов, ограничена. Все еще существует несколько открытых вопросов, особенно в том, что касается единицы анализа и процедуры сегментации, которой необходимо следовать. В исследовании De Smet M., et al., схема контент-

анализа была применена для анализа того, как онлайн-обучение у коллег (проводимое студентами четвертого курса) поддерживает асинхронные дискуссионные группы студентов первого курса [13]. Это исследование демонстрирует важную роль, которую репетиторство играет в онлайн-асинхронных дискуссиях, что принимается во внимание и является важным элементом модели.

С другой стороны, качественная информация также ценна для выполнения сложной задачи анализа взаимодействия и оценки вклада. Penashaff J. и Nicholls C. [27] использовали смешанный подход для анализа взаимодействия учащихся и построения смысла в обсуждениях на компьютерной доске объявлений. Количественный анализ использовался для изучения показателей участия и взаимодействия, а качественные процедуры использовались для анализа процессов формирования знаний и уточнения категориальной системы показателей и дескрипторов. Результаты показали, что студенты были вовлечены в процесс формирования знаний, который характеризовался главным образом прояснением, проработкой и интерпретацией. Более того, в исследовании Schrire S. [29] применяется объединение количественного анализа с качественной методологией для построения модели анализа совместного накопления знаний в асинхронном обсуждении. Модель позволяет рассматривать коммуникацию с различных точек зрения взаимодействия, познания и анализа дискурса. Контент-анализ дискурса был проведен на нескольких уровнях, сосредоточив внимание на самом дискуссионном форуме, темах обсуждения, сообщениях, а также на обмене сообщениями и перемещениях между ними. В результате удалось построить схему оценки накопления знаний в асинхронных дискуссионных группах. Схема объединяет интерактивные, когнитивные и дискурсивные аспекты CSCL. Аналогичным образом, Bratitsis T. и Dimitracopoulou A. [10] анализируют качество групповых взаимодействий при асинхронном обсуждении с помощью многоиндикаторной модели, основанной на количественных аспектах активного и пассивного поведения участников (т.е.

количестве написанных/отвеченных и прочитанных сообщений). Это может помочь преподавателю определить проблемные ситуации, возникающие в процессе обучения, а также определить индивидуальное поведение, которое может повлиять на сотрудничество, такое как пассивность и высокомерие.

В более поздней работе New K. и Cheung W., сообщают о качественном исследовании, изучающем методы фасилитации, используемые студентами-фасилитаторами для привлечения своих однокурсников к участию в асинхронных онлайн-дискуссиях [21]. Это исследование отличается от предыдущих в том смысле, что оно не фокусируется на роли преподавателя как фасилитатора и промоутера участия учащихся, но исследует фасилитацию со стороны сверстников. Чтобы выяснить, в какой степени участие студентов в онлайн-дискуссионном форуме является успешным, в исследовании рассматривается глубина тем обсуждения. Наконец, в нем сообщается о методах фасилитации, которые были продемонстрированы студентами-фасилитаторами. Однако простое рассмотрение глубины тем обсуждения само по себе не гарантирует качества обсуждения; публикации студентов могут быть просто продиктованы соображениями социализации и напрямую не связаны с разработкой учебных задач.

1.1.2 Управление данными о взаимодействии

Вся эта информация может быть легко собрана, автоматически обработана и проанализирована компьютерами в качестве источника количественных и качественных данных и представлена участникам для предоставления эффективной информации, например, о том, как все участники на самом деле выступают во время обсуждения, и о динамике каждого участника по отношению к группе. Следовательно, эффективное внедрение всей этой информации и извлеченных знаний в приложения CSCL создает основу для усиления поддержки, осведомленности и обратной связи, для достижения успешного процесса обсуждения в условиях совместной работы. Действительно, постоянная и быстрая обработка и представление этих количественных и качественных данных, а также их систематический анализ

на основе принципиальных показателей, которые измеряют тип и степень участия членов группы, могут положительно повлиять на мотивацию участников, эмоциональное состояние и способности решать проблемы и, как следствие, улучшить эффективность усвоения знаний. Конечная цель - извлечь релевантные знания о процессе обсуждения из всех возможных источников (например, активность пользователей, пассивность и эффективность; влияние участия; качественная оценка и т.д.). Следует обратить внимание, что в данном контексте информация относится к количественным и качественным данным, генерируемым обучающейся группой, тогда как знания относятся к результату обработки этой информации с помощью методов анализа и интерпретации. Эти знания будут возвращены и представлены членам учебной группы и ее наставнику для повышения осведомленности, обратной связи и создания каркаса.

1.2 Инженерное программное обеспечение CSCL-приложения

За последние годы на рынке появилось большое количество полнофункциональных систем электронного обучения, предлагающих дизайнерам и преподавателям адаптивные, мощные и удобные в использовании макеты для легкого и быстрого создания и организации курсов и мероприятий, который затем может быть настроен в соответствии с потребностями преподавателя, профилем учащихся и конкретными педагогическими целями. В этом разделе представлен обзор существующих систем электронного обучения для целей групповой деятельности. Затем дается обзор методов разработки программного обеспечения на основе повторного использования как для удовлетворения общих потребностей в области электронного обучения в целях повторного использования, так и для систематического удовлетворения текущих сложных и меняющихся требований в этой области.

1.2.1 Системы группового обучения

Программное обеспечение совместной работы — это парадигма программного обеспечения, предназначенная для облегчения сотрудничества между группами людей. Он был успешно применен к предметной области CSCL, ввиду острой потребности в предоставлении поддержки для облегчения накопления знаний в учебных средах. Для эффективной поддержки совместного обучения, системы программного обеспечения совместной работы включают в себя различные инструменты совместной работы, такие как хранилище документов, виртуальные классы, доски, чаты, тематические дискуссионные форумы и т.д. Кроме того легкая и расширяемая клиент-серверная архитектура позволяет реализовывать клиентские приложения для всех популярных вычислительных платформ и существующей базы пользователей, которая может исчисляться миллионами пользователей. В частности, тот факт, что веб-страницы представляют собой кроссплатформенный, единообразный, простой и интуитивно понятный интерфейс, определяемый браузером, приводит к тому, что веб-протоколы и приложения все чаще используются организациями для управления внутренним распространением информации, что способствует их расширению до функциональности программного обеспечения совместной работы.

Хотя большинство из этих систем программного обеспечения совместной работы поддерживают многие аспекты CSCL, они не полностью учитывают фундаментальные потребности пользователей в средах совместного обучения, такие как динамическая поддержка групповой осведомленности, специальные компоненты для управления осведомленностью и взаимодействие между различными приложениями для поддержки совместной работы. Более того, большинство из них не предоставляют ни обработки лог-файлов, ни инструментов для анализа обработанной информации, а регистрируют взаимодействие в виде разрозненного огромного объема плохо структурированной информации с высокой степенью избыточности, что требует эффективной системы

обработки данных для анализа этой сложной информации. Наконец, не проводится никакого анализа реальных навыков и намерений учащихся на основе обоснования конкретной концептуальной модели анализа данных и управления ими, и в результате возможности информирования и обратной связи сильно ограничены.

Системы управления обучением (СУО) — это программные пакеты, позволяющие управлять образовательным контентом, а также интегрирующие инструменты, которые поддерживают большинство потребностей группового программного обеспечения, такие как электронная почта, дискуссионные форумы, чат и виртуальные классы. За последние годы на рынке появилось большое количество полнофункциональных веб-систем СУО, предлагающих дизайнерам и преподавателям универсальные, мощные и удобные в использовании макеты для простого и быстрого создания и организации курсов и мероприятий, которые затем могут быть дополнены конкретными материалами курса и адаптированы к потребностям пользователя, инструктора и педагогических целей. Большинство из этих систем используют преимущества таких важных стандартов, как инициатива открытых знаний (ОКИ) и глобальный учебный консорциум (IMS), чтобы обеспечить устойчивую совместимость с разнородными системами в области электронного обучения.

Наконец, системы управления контентом с открытым исходным кодом (CMS) рассматриваются как абстракция или надмножество СУО и становятся очень популярными при создании веб-систем обучения. Действительно, гибкость этих систем и мощная поддержка со стороны сообщества разработчиков с открытым исходным кодом могут сделать CMS лучшей альтернативой СУО. В контексте этого тезиса, системы СУО, безусловно, слишком ограничены и налагают множество барьеров, когда речь заходит о включении непредвиденного аспекта в их внутренний дизайн, такого как предлагаемая концептуальная модель информации и знаний. Системы CMS же, задумывались не только для целей обучения, и поэтому они открыты для

создания очень гибких систем обучения.

Несмотря на большую поддержку систем СУО в таких важных областях, как коммуникация, сотрудничество и оценка, в целом оказывается незначительная поддержка осведомленности и обратной связи, что имеет основополагающее значение в данном контексте. Несмотря на то, что исходный код некоторых систем СУО может быть доступен, адаптация их внутренних компонентов для включения показателей новостей, которые будут собираться в результате взаимодействия участников, не поддерживается, равно как и представление участникам в подходящих форматах полученных знаний в персонализированной, динамичной форме с переключением на осведомленность, обратную связь, мониторинг и т. д. Другим распространенным недостатком является отсутствие интероперабельности, что делает приложения зависимыми от языка программирования, базовой инфраструктуры и так далее.

Две платформы СУО особенно хорошо зарекомендовали себя на рынке и активно внедряются образовательными организациями. Moodle вместе с проектом Sakai являются крупнейшими движениями с открытым исходным кодом, увеличивающими свою долю в образовательном пространстве. Moodle разработан с использованием конструктивизма, чтобы помочь преподавателям создавать эффективные сообщества онлайн-обучения, в то время как Sakai — это масштабная разработка программного обеспечения с открытым исходным кодом для проектирования, создания и развертывания новых сред совместного обучения для высшего образования. Это позволяет учебным заведениям максимально адаптировать Sakai в соответствии со своими педагогическими потребностями и технологическими требованиями.

Несмотря на предоставление обширного API, в настоящее время как в системах Moodle, так и в системах Sakai отсутствует надлежащая документация, и адаптация к различным потребностям важных модулей, таких как компонент ведения журнала, является довольно хаотичной. Кроме того, поскольку Moodle полностью написан на PHP, а Sakai - на Java, это создает

проблемы с функциональной совместимостью, которые могут повлиять на решение о выборе технологической платформы, под которой он работает. Чтобы преодолеть эти проблемы, оба проекта разрабатывают API веб-сервиса, но пока был создан только набор методов управления пользователями и курсами, а также администрирования в целях тестирования.

Особым случаем также является система LRN, которая предоставляет специальный модуль отслеживания пользователей для полной поддержки мониторинга и информирования во время совместной работы, так что можно представить множество мнений в виде статистики о том, что происходит в каждом рабочем пространстве, сайте и сообществе. Несмотря на такую широкую поддержку и большой объем собранной информации, не существует ни конкретных показателей для классификации и измерения реальных навыков и намерений, ни процесса непрерывной обработки и анализа этой информации, полученной эффективным образом, в целях повышения осведомленности и обратной связи.

В целом, если мы спросим себя, как управляются пользователи и ресурсы из разных систем CVO и CSCL и какие индивидуальные функции имеют значение при переходе с одной системы на другую, мы увидим, что их реализация и, самое главное, их реализация с использованием конкретных языков программирования и платформ сильно отличаются. Тем не менее, требования и дизайн всех этих систем не так уж сильно отличаются друг от друга, а это означает, что между всеми ними существует потенциально важная общая основа, которая позволяет избежать разработки новых систем с нуля. Далее предлагается общий подход, основанный на повторном использовании, который следует этим принципам.

1.2.2 Методы разработки программного обеспечения

В этом подразделе анализируются несколько методов разработки программного обеспечения с точки зрения общих архитектурных решений для систем электронного обучения. Эти методы описаны в исторической перспективе, от самых традиционных до новейших. Также обсуждается

определенный важный вклад в область электронного обучения с использованием этих методов.

Компоненты были одним из самых успешных ресурсов в области повторного использования программного обеспечения. В данном методе описываются все действия, в контексте полного жизненного цикла программного обеспечения на основе компонентов. Программный компонент имеет определенные контрактные интерфейсы и явные контекстные зависимости, компонент может быть развернут независимо и может быть составлен третьими сторонами. Однако в работе Dimitriadis Y., et al., применение компонентов при разработке приложений CSCL показывает, что возможности повторного использования, достигаемые компонентами, явно низки из-за их ограниченной области применения [14]. В этих подходах также отсутствуют важнейшие аспекты, лежащие в основе парадигмы CSCL, такие как анализ данных взаимодействия и представление извлеченных знаний.

Anido-Rifon L., et al., предлагают трехуровневую компонентную структуру, ориентированную на интерактивные и совместные образовательные приложения [6]. При таком подходе события, создаваемые данным компонентом, обрабатываются и доставляются удаленным компонентам в том же приложении. Другой компонент позволяет совместно использовать и распространять события, выполняемые в пользовательском интерфейсе общего приложения, где действия пользователя будут перенаправляться каждому другому пользователю в группе. Хотя предлагаемая структура поддерживает большинство сценариев с учетом их особенностей, она не фокусируется ни на обработке и анализе информации о событиях, полученной в результате действий пользователя, ни на том, как представить эту информацию пользователям с точки зрения динамической поддержки групповой осведомленности, конкретных компонентов для информирования и управления обратной связью.

В работе Vaselo V. и Becker K. [8], представлен интересный подход к компонентной архитектуре для поддержки разработки приложений для

совместной работы, особенно рассматриваются аспекты коммуникации, сотрудничества и обмена информацией. Однако такие аспекты, как осведомленность и управление знаниями, опущены и должны быть приняты во внимание. Для целей этого тезиса стоит упомянуть здесь подходы, которые были использованы в работе Vote Lorenzo M., et al., которые указывают на использование распределенных вычислительных сред при разработке компонентов для областей совместного обучения [9]. В контексте этого исследования распределенные вычисления также используются для конкретных приложений совместного обучения, чтобы высокоэффективно удовлетворять многие важные потребности этих приложений, такие как анализ данных о групповой деятельности и управление ими. Например, grid-вычисления предлагают вычисления с высокой пропускной способностью и интенсивным использованием данных, что значительно облегчает процесс внедрения информации и знаний в эти приложения, позволяя предоставлять пользователям постоянную осведомленность в режиме реального времени и обратную связь.

Marques J. и Navarro L. также предлагают распределенную и децентрализованную инфраструктуру, целью которой является поддержка распределенного группового обучения и командной работы [23]. Эта инфраструктура основана на механизмах распространения событий, обеспечивающих осведомленность, чтобы участники могли получать уведомления и, таким образом, быть осведомленными о прогрессе групп, к которым они принадлежат. В нем описывается, как собирать и распространять события таким образом, чтобы уведомлять участников группы о действиях других людей. Однако в нем не описывается, как эта информация должна быть представлена участникам, и, таким образом, теряется некоторая эффективность в обеспечении осведомленности.

Несмотря на то, что все эти подходы оказались успешными для создания эффективного группового программного обеспечения и, в частности, приложений CSCL, необходимы новые парадигмы и методы разработки

программного обеспечения для удовлетворения новых сложных педагогических и технологических требований, возникающих в области CSCL, при одновременном сокращении общих инженерных усилий, необходимых для их удовлетворения. С этой целью обычно разрабатываются универсальные платформы и фреймворки для создания сложных программных систем с использованием метода повторного использования, фокусируясь на системах семейства, а не на отдельных системах. Этот подход был успешно применен в различных областях, что позволяет создавать приложения повышенного качества, сокращая как затраты, так и время разработки. С этой целью используется парадигма универсального программирования (УП). Вышеописанная парадигма появилась в контексте разработки компьютерного программного обеспечения как инновационная концепция, которая пытается сделать программное обеспечение настолько общим, насколько это возможно, без потери эффективности. УП достигает своей цели, выделяя взаимосвязанное семейство высокого уровня из общего набора требований. Благодаря применению этого метода, особенно на этапах проектирования, разрабатывается программное обеспечение, предлагающее высокую степень абстракции, которая применима к широкому спектру ситуаций и областей.

Более того, парадигма разработки на основе моделей (MDD) и поддерживающая ее платформа, а именно, архитектура, управляемая моделью (MDA), в последнее время привлекает большое внимание, поскольку она позволяет разработчикам программного обеспечения и организациям охватывать каждый важный аспект программной системы с помощью соответствующих моделей. MDA, в свою очередь, представляет собой новую методологию разработки программного обеспечения, которая доказала свою эффективность в продвижении возможности повторного использования, предоставляя необходимые инструменты для полного использования возможностей повторного использования программного обеспечения. Главная идея MDA заключается в том, чтобы указать уникальное и независимое

представление (называемое платформенно-независимой моделью) программного артефакта и автоматически с помощью генератора создать несколько реализаций (называемых платформенно-специфичной моделью). MDA предоставляет большие преимущества с точки зрения полной поддержки разработки всего цикла, снижения затрат, качества программного обеспечения, возможности повторного использования, независимости от технологии, интеграции с существующими системами, масштабируемости и надежности, гибкой эволюции программного обеспечения и стандартизации, поскольку поддерживается группой управления объектами (OMG). MDD и MDA сильно полагаются на несколько методов моделирования, таких как анализ предметной области, что включает в себя процесс анализа связанных программных систем в предметной области с целью нахождения их общих и переменных частей. Обычный результирующий продукт представляет собой модель предметной области в виде общей архитектуры, которая описывает все системы в приложении.

Существует множество мнений о том, что такое MDA, а что нет. Однако OMG, как наиболее авторитетная точка зрения, фокусирует MDA на главном видении: позволить разработчикам создавать приложения независимо от конкретных платформ реализации (таких как язык программирования или промежуточное программное обеспечение). С этой целью OMG предлагает следующие принципы для разработки MDA: во-первых, разработка независимой от платформы модели на основе UML (PIM), во-вторых, одна или несколько моделей, которые являются зависимыми от платформы моделями (PSM). Наконец, для отображения из PIM в PSM необходима определенная степень автоматизации с помощью описаний.

Laforcade P., et al., делают обширный обзор из литературы обещаний, вызовов и проблем, возникающих при применении принципов и теорий, использующих парадигмы MDD/MDA для моделирования и развертывания систем электронного обучения [22]. Они приходят к выводу, что такой подход к разработке программного обеспечения может помочь разработчикам

сократить разрыв между конкретными требованиями к инструкции и программными архитектурами, которые поддерживают реализацию, время выполнения и регулирование этой инструкции.

Наконец, для целей этого тезиса важно упомянуть здесь подход сервис-ориентированной архитектуры (SOA), который представляет собой следующий шаг в разработке программного обеспечения, помогающий организациям удовлетворять их все более сложный набор потребностей и задач, особенно в распределенных системах. SOA приносит пользу разработке программного обеспечения главным образом за счет динамического обнаружения и вызова соответствующих служб для выполнения запроса из разнородных сред, независимо от деталей и различий этих сред. Делая сервис независимым от контекста, SOA предоставляет программному обеспечению важные нефункциональные возможности для распределенных сред (такие как масштабируемость, гетерогенность и открытость) и значительно упрощает процессы интеграции.

Несмотря на то, что SOA может быть реализована с помощью других технологий, за последние несколько лет веб-сервисы стали играть главную роль в SOA, предоставляя набор стандартных протоколов, которые отвечают основным потребностям SOA. Использование широко распространенных протоколов и стандартов, таких как XML, UDDI, WSDL, SOAP и HTTP, представляет собой краеугольный камень подхода к веб-сервисам, который обеспечивает подходящую технологию для реализации ключевых требований SOA. Это связано с тем, что эти протоколы позволяют сервису быть независимым от платформы и языка, динамически размещаться и вызываться, а также взаимодействовать в сетях различных организаций. В результате технология веб-сервисов обеспечивает более низкие затраты на интеграцию наряду с гибкостью и упрощением настройки. Далее приводится краткое описание основных протоколов:

- простой протокол доступа к объектам (SOAP): это облегченный механизм, используемый для обмена XML-сообщениями между

приложениями независимо от языка программирования или операционной системы. Это определяет формат сообщения, которое будет передаваться между приложениями, обычно (но не исключительно) по протоколу HTTP;

- язык описания веб-сервиса (WSDL): обеспечивает необходимую структуру для описания функциональных возможностей веб-сервиса посредством предоставления грамматики XML для использования в веб-сервисе;
- универсальное описание, обнаружение и интеграция (UDDI): Эта спецификация предоставляет структуру, используемую для описания и регистрации определенной логики процесса, с помощью которой клиентские приложения могут узнавать о предлагаемых услугах и их местоположении или адресах.

Таким образом, веб-сервисы, основанные на стандарте eXtensible Markup Language (XML), обеспечивают совместимость различных приложений, работающих на разнородных платформах, позволяя автоматизировать процессы в различных приложениях. Абстракция модели веб-сервисов, включающая вышеупомянутые протоколы, показана на рисунке 1. В основе мы находим протокол связи (TCP/IP). Параллельно стеку веб-сервисов могут быть подключены различные протоколы, чтобы гарантировать надлежащую практику в отношении безопасности, качества услуг и управления.

Несмотря на то, что веб-сервисы были продемонстрированы как идеальная технология для внедрения SOA, некоторые недостатки и подводные камни, такие как проблемы с производительностью и несовместимость безопасности с другими технологиями, нуждаются в тщательном анализе и устранении.



Рисунок 1 - Структура веб-сервиса

В большинстве исследований, используются преимущества сервис-ориентированного подхода для улучшения разработки программных систем. Одним из показательных примеров применения этого подхода в области электронного обучения является работа Gutl С., который представляет открытую, распределенную сервис-ориентированную архитектуру для гибкого преподавания и учебной деятельности [18]. В этом исследовании он обсуждает накопленный опыт и показывает, как система на основе SOA, могла бы достичь важных целей исследований и разработок в этом контексте, таких как поддержка различных парадигм преподавания и обучения, персонализация при поиске, управлении и представлении соответствующей информации, необходимой для выполнения учебной деятельности, и адаптация пользователя как для системы, так и для процесса обучения за счет улучшения знаний о поведении пользователя. Однако в этом исследовании даже не упоминаются другие не менее важные преимущества использования сервис-ориентированного подхода к обучению, такие как надежность, производительность и совместимость с существующими и унаследованными системами.

Этот тезис продвигает все эти подходы еще на один шаг вперед, предоставляя методы, основанные на новом поколении, которые используют

высокоуровневую спецификацию моделей предметной области CSCL и обеспечивают ее реализацию с помощью полуавтоматического генератора. Пользователи генератора увидят систему, которая позволяет им перейти от модели спецификации к другой модели реализации без необходимости разбираться во внутренних деталях генератора.

1.3 Распределенные и grid-вычисления для удовлетворения современных потребностей в CSCL

Согласно работе Foster I, et al., grid-вычисления появились как способ воплощения концепции сетевой вычислительной системы, которая обеспечивает широкий доступ не только к массивным информационным ресурсам, но и к массивным вычислительным ресурсам в целом [15]. Концепция вычислительной сети берет свое начало в широкомасштабных распределенных вычислениях и распространяется на крупномасштабное, гибкое, безопасное, скоординированное совместное использование ресурсов между динамическими наборами отдельных лиц, учреждений и ресурсов. Grid-архитектура, представлена в виде пяти уровней, которые могут быть распределены на разных уровнях (рисунок 2).

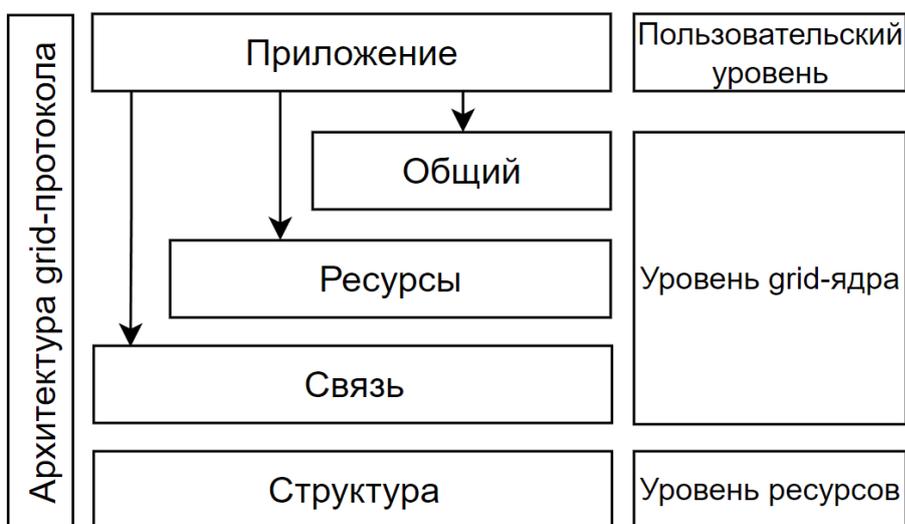


Рисунок 2 - Многоуровневая grid-архитектура

За последние годы появилось довольно много сетей электронного обучения, которые используют архитектуры grid для предоставления более мощных ресурсов образовательным организациям. Данное решение позволяет исполнять важные нефункциональные требования, возникшие в этом контексте, такие как масштабируемость, доступность и распределение вычислительной мощности, а также как возможность хранения большого объема данных.

Благодаря такому подходу grid обеспечивает идеальный контекст для поддержки и получения основных преимуществ для приложений CSCL. К таким важным особенностям относятся: большой масштаб сетевых инфраструктур, широкое географическое распределение ресурсов, возможность администрирования из разных организаций, прозрачный и надежный доступ, а также возможность предоставления доступа к общим, разнородным ресурсам в очень динамичных средах. Учитывая эти преимущества, предоставляемые grid, образовательные организации могут использовать настоящие среды совместного обучения, которые позволяют привлекать большое количество одиночных / групповых участников, которые потенциально могут принадлежать ко многим различным организациям, возможно, расположенным в самых разных местах, и прозрачно использовать огромное разнообразие как программного, так и аппаратного обеспечения при одновременном улучшении взаимодействия между людьми.

Таким образом, использование потенциала производительности, присущего grid-инфраструктуре для приложений CSCL, позволяет значительно улучшить условия совместной работы. Более того, сочетание с другими технологиями, такими как архитектуры, ориентированные на обслуживание, и веб-сервисы, позволяет разработчикам решать важные проблемы CSCL, такие как интеграция, интероперабельность, отказоустойчивость и гибкость, чтобы удовлетворять потребности различных, гетерогенных и унаследованных сред. В этом контексте, становится очевидной ключевая роль, которую играют архитектуры, основанные на SOA,

и, в частности, парадигма веб-сервисов. Технологии веб-сервисов обеспечивают как интероперабельность для преодоления большой сложности промежуточного программного обеспечения grid, так и простоту управления и предоставления разнородного, сложного учебного контента и курсов.

Тем не менее, по-прежнему ощущается острая нехватка поддержки для конкретных ситуаций и поддоменов электронного обучения, таких как CSCL. Всего несколько систем grid предназначены специально для поддержки CSCL, и в этом контексте требуется большее изучение grid-архитектур, учитывая присущий им потенциал с точки зрения обмена огромными объемами обучающих данных и ресурсов во время совместной работы, затрачивая разумное количество ресурсов. Несмотря на то, что ранее рассмотренные области исследований были в значительной степени изучены, их анализ показал, что они все еще далеки от конечного результата и все еще требуется много усилий для преодоления проблем, которые являются частью самого их обоснования.

В данном разделе был проведен анализ литературы по теме применяемых решений в приложениях CSCL. В результате было выявлено, что приложения CSCL, в основе своей, имеют проблему в узконаправленности предлагаемых решений. На основе анализа литературы было выяснено, что унификация процесса сбора, обработки, анализа и представления информации и знаний, полученных в результате групповой деятельности, может решить проблему узконаправленности решений. Таким образом, для решения проблем узкого направления решений, необходимо предоставить улучшенный метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL, с внесением структуры эффективного управления и анализа данных взаимодействия, генерируемых в ходе учебной практики.

2 Проведение анализа существующих решений в рамках рассматриваемой тематики

В рамках предварительного анализа существующих решений, была проведена экспертиза существующих решений по поддержке преподавателей.

На основе проведенного анализа систем, в рамках данной работы, теме, можно выявить закономерность в архитектуре и структуре предлагаемых сторонними разработчиками подходов к реализации предлагаемых ими решений, что в последствии можно использовать при разработке собственной системы, подчеркивая достоинства и нивелируя недостатки других решений.

2.1 Проведение анализа работы существующих решений различных площадок

Для решения данной задачи были выбраны некоторые системы, которые функционируют в различных школах и университетах Европы. Стоит отметить, что данная работа фокусируется на анализе систем и инструментов анализа взаимодействия между участниками (преподавателями, учениками и т.д.), поскольку оценить ту или иную систему в целом, с точки зрения эффективности является несколько затруднительным, поскольку результаты эффективности системы в целом, сильно зависимы от окружения (как человеческого, так и материального).

Инструменты и системы анализа взаимодействия, анализируют взаимодействия между участниками в онлайн-средах совместного обучения с использованием компьютерных технологий, пытаясь использовать вспомогательные механизмы. Существует большое разнообразие подходов к использованию информации, полученной в результате анализа взаимодействий между участниками. Одним из возможных объяснений такого разнообразия может быть различная теоретическая точка зрения, на которой основана каждая система. Однако даже системы, которые придерживаются

одних и тех же теоретических взглядов на обучение, используют разные стратегии, когда дело доходит до педагогического вмешательства [11]. Эти системы используются для поддержки студентов и преподавателей в их задачах в онлайн-образовательных средах, где они представлены в соответствии с их отношением к учебной среде (платформе), в которой анализируются данные взаимодействия, в соответствии с участником, к которому обращается анализ, к типу источника данных взаимодействия, а именно, каждый инструмент взаимодействия, который использует для анализа (входные данные), способ, которым инструмент выполняет анализ собранных данных (используемые методы) и, наконец, цель анализа (ожидаемая функция).

Чтобы быть более конкретным, статус, который присваивается фактическим инструментам анализа взаимодействия, подразделяется на несколько категорий в зависимости от их отношения к среде обучения, в которой анализируются данные взаимодействия. Большинство инструментов анализа взаимодействия служат компонентами веб-сред обучения, собирая данные из образовательных инструментов, таких как платформы для ведения блогов, форумы и вики-страницы. С другой стороны, модульная объектно-ориентированная динамическая среда обучения является наиболее популярной платформой системы управления обучением для извлечения данных, в то время как для некоторых из перечисленных инструментов анализа взаимодействия, данные получаются с платформ массовых, открытых онлайн-курсов или других персональных сред обучения [16]. В целом, большинство инструментов анализа взаимодействия, используют данные с одной платформы, в то время как, лишь несколько инструментов используют данные с разных платформ, которые они впоследствии объединяют.

В результате анализа инструментов анализа взаимодействия, были выявлены три типа целевых пользователей, а именно студенты, преподаватели и исследователи. Преподаватели и студенты, несомненно, являются основными пользователями представленных инструментов анализа

взаимодействия, хотя в некоторой степени присутствуют и исследователи. Что касается поддержки коллег в асинхронных средах совместного веб-обучения, основное наблюдение о существующих инструментах и системах анализа взаимодействия заключается в том, что лишь немногие из них предназначены для всех типов пользователей. На самом деле, большинство инструментов предназначены исключительно для студентов или преподавателей, в то время как обе группы могут получить доступ лишь к небольшому количеству инструментов. Кроме того, учащимся обычно предоставляется ограниченный доступ к инструментам анализа взаимодействия из-за их ограниченных прав пользователя, а также их роли в процессе обучения. Таким образом, учащимся обычно оказывается меньшая поддержка в визуализации и анализе их совместной учебной деятельности в режиме онлайн, и даже в этих случаях рассматриваемые инструменты не разработаны с учетом самоконтроля учащихся.

Что касается используемых источников данных, то визуализируемые учебные артефакты, по большей части, создаются или используются учащимися (например, анализ их содержания). Эти артефакты включают сообщения в блогах или форумах, ответы на вопросы, запросы о помощи или, в более общем плане, ресурсы, созданные учащимися, которые, в свою очередь, могут быть общедоступными документами или очень специфическими артефактами, такими как веб-сайты или программный код. Социальное взаимодействие (включая журналы, используемые для отслеживания компьютерной активности пользователей) занимает второе место среди наиболее часто отслеживаемых и визуализируемых элементов, и примеры включают анализ комментариев к сообщениям в блогах и форумах или сообщений в чате, которыми обмениваются учащиеся, другие формы взаимодействия и так далее. Многие инструменты анализа взаимодействия также отслеживают использование ресурсов, либо для того, чтобы помочь оценить вовлеченность, либо для информирования учителей о том, в какой степени ресурсы используются их учениками – другими словами, либо для

определения показателей того, насколько ресурсы считаются релевантными, либо для оценки прикладываемых усилий учащимися. Время, затраченное на занятие, также учитывается этими инструментами, и это помогает учителям выявлять учащихся, которые потенциально имеют проблемы с усвоением материала. Более того, затраченное время иногда визуализируется, чтобы учащиеся могли сравнить усилия, которые они вкладывают в ту или иную деятельность, с усилиями своих сверстников. Кроме того, существуют результаты тестирования и самооценки, которые относятся к уровню знаний учащихся, чтобы они могли лучше понять прогресс в обучении (по сравнению с пониманием, получаемым при учете компонента «затраченное время»). Наконец, существуют некоторые инструменты анализа взаимодействия, которые не указывают источники данных, которые они используют. Что касается используемых методов анализа взаимодействия, то есть способа, которым инструмент анализа взаимодействия проводит анализ собранных данных, большинство инструментов анализа взаимодействия используют методы интеллектуального анализа образовательных данных, чтобы понять, как учащиеся учатся, и определить любые аспекты, которые могут улучшить обучение, а также образовательные аспекты. В связи с этим визуализация информации является вторым наиболее часто используемым методом анализа взаимодействия, хотя со стороны пользователей есть положительные реакции на представленную им визуализированную информацию, данные исследований об этом методе ограничены, особенно с точки зрения оценки эффективности, полезности, удобства использования, а также точный способ, которым пользователи воспринимают и обрабатывают эту информацию.

Что касается различных типов визуализации индикаторов анализа взаимодействия, то наиболее популярными представлениями являются гистограммы, линейные графики, таблицы и круговые диаграммы, которыми пользуются все группы пользователей. Аналогичным образом, используемые типы визуализации существенно не различаются в разных образовательных учреждениях. Другая группа инструментов анализа взаимодействия,

описанных в этом разделе, применяет формальные концепции и методы анализа социальных сетей для изучения и отображения структуры групповой активности. Кроме того, существует ряд инструментов анализа взаимодействия, использующих подходы, связанные с анализом контента. Наконец, существует несколько инструментов анализа взаимодействия, которые основаны на несоответствиях между собранными данными и стандартами оптимального совместного функционирования для руководства или внешнего регулирования процессов сотрудничества и коммуникации. Используя искусственный интеллект в образовании, эти инструменты анализа взаимодействия используют сложные вычислительные алгоритмы для интерпретации данных и мониторинга совместной работы, чтобы анализировать, а также «диагностировать» взаимодействие учащихся и, следовательно, делать выводы или делать выводы о том, где учащиеся могут столкнуться с трудностями.

Что касается цели анализа (ожидаемой функции), существуют инструменты анализа взаимодействия, которые помогают учащимся в самоконтроле их собственных взаимодействий, а также взаимодействий их коллег, чтобы способствовать осознанию, метапознанию и, следовательно, саморегуляции их обучения в рамках общей деятельности и процесса. С другой стороны, существует множество инструментов, адресованных учителям, особенно в веб-контекстах, поддерживающих их усилия по мониторингу некоторой структурированной и компактной информации о том, что происходит во время учебной деятельности. Такие инструменты позволяют учителям принимать более адекватные и оперативные меры, связанные с качеством результата деятельности и/или качеством самого сотрудничества. Ряд инструментов предназначен для изучения социальных и качественных параметров или измерения качества сотрудничества. Эти инструменты адресованы исследователям и разработчикам учебных мероприятий, оказывая им поддержку в их усилиях по разработке стратегий обучения. Наконец, существуют руководящие инструменты анализа

взаимодействия, которые консультируют участников о способах регулирования их действий и улучшения их собственного вклада и сотрудничества со своими коллегами в целом.

Тем не менее, стоит отметить, что большинство инструментов анализа взаимодействия не предоставляют подробного списка используемых показателей анализа взаимодействия [20]. Большинство индикаторов анализа взаимодействия этих инструментов создаются в автономном режиме и неавтоматическим способом, обычно через запланированные промежутки времени (на основе только пост-анализа следов взаимодействия), без подробного отчета о действиях учащихся и без учета точки зрения учащихся (поскольку это связано, например, с целями или намерениями их саморегулирующихся действий). Следует также отметить, что большинство инструментов и систем анализа взаимодействия, описанных в данном разделе, являются прототипами исследований, которые, как правило, фокусируются на конкретном исследовательском вопросе и должны рассматриваться с точки зрения этого вопроса. В этом ключе большинство инструментов анализа взаимодействия подвергаются минимальной настройке и адаптации настроек, например, в отношении структуры и стиля действия, или не предлагают настройки или выбора опций в отношении доступных инструментов каждый раз, или любой другой формы настройки профиля пользователя и т.д. Еще один момент, который стоит упомянуть, заключается в том, что большинство инструментов анализа взаимодействия ориентированы на университетские условия, в отличие от условий среднего или начального образования.

Для наглядности полученных в результате анализа данных, их необходимо структурировать.

2.2 Результаты сравнительного анализа существующих решений

В данном разделе в виде таблицы представлены результаты анализа существующих решений. В таблице 1 представлены результаты анализа типа

платформ, пользователей и источника данных.

Таблица 1 - Краткое описание типов платформ, анализируемых пользователей и анализируемых источников данных

Платформы	Тип платформы					Анализируемые пользователи			Источник данных				
	Standalone-приложение	Компонент системы	Web-приложение	Система управления обучением/контентом	Мобильное	Студенты	Преподаватели	Исследователи	Артефакты	Социальное взаимодействие	Используемые ресурсы	Затраченное время	Результаты тестов/занятий
i-Bee	+					+			+		+		
ForumExplor	+							+	+				
CONTRAS	+					+			+		+		
Web2SRL			+			+			+		+		
PELS			+			+			+		+		
DIAS			+			+	+	+	+	+	+	+	
RedCoMulApp			+			+	+		+		+	+	+
ELLI component			+			+	+		+	+	+	+	
TrAVis			+			+	+		+	+	+	+	
VisCa			+			+	+				+	+	
Course signals	+		+			+						+	+
OLI Dashboard			+				+		+		+	+	+
SAM			+			+	+		+		+	+	
CACVOystem			+			+	+		+		+	+	+
AulaNet			+			+	+		+		+		+
Analytic Tool		+					+	+	+		+		
Argunaut	+						+		+		+		
iHelp			+				+		+	+			
eLAT	+						+		+		+	+	
LOCO-Analyst			+	+			+		+	+	+	+	+
GLASS			+			+	+		+		+	+	+
Student Inspector			+				+		+		+	+	
GISMO				+			+		+	+	+	+	
MooDog				+		+	+		+		+		
LeMo				+			+	+	+		+	+	
MATEP			+			+	+		+		+	+	
Moodle Dashboard				+			+		+	+	+	+	+

Продолжение таблицы 1

Платформы	Тип платформы					Анализируемые пользователи			Источник данных				
	Standalone-приложение	Компонент системы	Web-приложение	Система управления обучением/контентом	Мобильное	Студенты	Преподаватели	Исследователи	Артефакты	Социальное взаимодействие	Используемые ресурсы	Заграченное время	Результаты тестов/занятий
iTree	+				+	+			+	+	+		
LA e-Rubric				+			+		+		+	+	+
Awariness Dashboard					+	+	+		+	+	+		+
MTDashboard					+	+	+		+		+		
SNAPP			+			+	+			+		+	
TUT Circle Dashboard			+			+				+			
Classroom View			+				+			+		+	
TeacherADVisor			+				+	+		+	+		+
Omega+			+			+	+	+	+		+	+	
MentorChat	+					+	+		+	+	+		
CALICO			+				+	+	+		+	+	
Cohere			+					+	+	+			

Исходя из полученных результатов, можно сделать вывод, что в большинстве своем, учебные заведения отдают предпочтение web-приложениям, в которых анализируются преподаватели и ученики, используя в качестве входных данных артефакты и используемые ресурсы.

В таблице 2 представлены результаты анализа обработки данных, полученных в результате учебной деятельности.

Таблица 2 – Краткое описание используемых методов анализа данных

Платформы	Техника анализа					Ожидаемая функция								
	Интеллектуальный анализ	Анализ социальных сетей	Анализ материала	Визуализация информации	Искусственный	Мониторинг	Участие	Предсказание	Вмешательство	Саморегулирование	Оценка студентов	Персонализация	Рекомендация	Размышление/Осознание
i-Bee	+	+	+	+		+	+			+		+		+
ForumExplor	+		+	+		+			+					+
CONTRAS	+					+	+			+		+		+
Web2SRL	+			+		+	+			+				+
PELS	+			+		+	+			+				+
DIAS	+	+		+		+	+		+	+		+		+
RedCoMulApp	+			+		+	+		+	+		+		+
TrAVis	+	+		+		+	+		+	+	+	+		+
VisCa	+			+		+	+		+	+		+		+
Course signals	+			+	+	+	+	+		+		+		+
OLI Dashboard	+			+		+		+	+		+			+
SAM	+			+		+	+	+	+			+		+
CACYOystem	+			+		+	+		+	+	+	+		+
AulaNet	+		+	+		+			+		+			+
Analytic Tool	+			+		+	+		+	+				+
Argunaut	+		+	+		+			+		+			+
iHelp	+	+		+		+			+					+
eLAT	+			+		+			+		+			+
LOCO-Analyst	+	+	+	+		+	+	+	+		+	+		+
GLASS	+		+	+		+	+	+	+		+	+		+
Student Inspector	+			+		+		+	+		+			+
GISMO	+	+		+		+			+			+		+
MooDog	+			+		+	+		+	+				+
LeMo	+			+		+			+					+
MATEP	+			+		+	+		+	+				+
Moodle Dashboard	+	+		+		+			+		+			+
LA e-Rubric	+	+		+		+			+			+		+
Awariness Dashboard	+	+	+	+		+	+		+	+	+			+
iTree	+			+		+	+			+		+		+
StepUp!	+	+	+	+		+	+	+	+	+		+		+
MTDashboard	+			+		+	+		+	+		+		+

Продолжение таблицы 2

Платформы	Техника анализа					Ожидаемая функция								
	Интеллектуальный анализ	Анализ социальных сетей	Анализ материала	Визуализация информации	Искусственный	Мониторинг	Участие	Предсказание	Вмешательство	Саморегулирование	Оценка студентов	Персонализация	Рекомендация	Размышление/Осознание
SNAPP	+	+		+		+	+	+	+	+				+
TUT Circle Dashboard	+	+		+		+	+			+		+		+

Результаты следующего анализа гласят, что чаще всего, интерпретация полученных данных происходит за счет использования технологий интеллектуального анализа, а также визуализации информации. Зачастую, от системы ожидают проведение мониторинга, участие и вмешательство в учебный процесс, а также размышление об эффективности предлагаемых стратегий. На основании полученных результатов, мы можем выявить тенденции в популярности/нужде в системах определенного типа, что позволит сосредоточиться на требуемых от системы аспектах.

В данном разделе диссертации была решена задача идентификации проблем систем CSCL. Общие тенденции, которые были определены в таблице 1 и 2, показывают заинтересованность в определенных решениях.

Обозначенные проблемы можно решить с помощью улучшения метода разработки программного обеспечения на основе повторного использования компонентов приложений CSCL, с внесением структуры эффективного управления и анализа данных взаимодействия, генерируемых в ходе учебной практики.

Рассмотрение и предоставление решения вышеописанных проблем будут представлены в следующем разделе магистерской диссертации.

3 Вычислительная модель CSCL на основе повторного использования

Разработка четкой и хорошо структурированной концептуальной модели, представляет собой принципиальный способ проектирования вычислительной модели, которая реализует процесс встраивания информации и знаний в приложение CSCL. Задача заключается в предоставлении инновационного и эффективного механизма, который структурирует и классифицирует информацию в рамках процессов сотрудничества высокого уровня, в то время как он определяет потенциальные показатели среднего и низкого уровня, которые измеряют и оценивают каждый процесс. Этот механизм облегчает создание переносимой, общей и многоразовой онтологии совместного обучения для представления, изучения и вывода знаний о каждом процессе совместной работы. Это позволяет разработать эффективную вычислительную модель, которая максимально точно отражает и описывает выполнение задач, индивидуальное и групповое поведение, динамику взаимодействия, взаимоотношения участников и групповую поддержку. Кроме того, эта универсальная, надежная и многоразовая платформа может использоваться для систематического создания приложений CSCL, наделенных расширенными возможностями для обеспечения более эффективного управления знаниями.

3.1 Универсальное программирование

В традиционных формах инжиниринга производительность, качество и стоимость являются настолько важными факторами в производственных процессах, что от них зависит само выживание компаний. По этой причине прилагаются огромные усилия для улучшения техник, методов и инструментов, которые доступны для разработки продукта. Однако в случае разработки программного обеспечения до сих пор не было достигнуто

аналогичного прогресса, хотя существенной разницы с другими формами инжиниринга нет. Действительно, тот факт, что технология более новая, а продукт очень сложный, может объяснить причины, но ключ, несомненно, следует искать в том факте, что концепция возможности повторного использования использовалась недостаточно [26].

Повторное использование ранее созданных деталей продукта приводит к снижению затрат и повышению производительности и качества до такой степени, что промышленные процессы совершат большой скачок вперед. Во всех передовых формах инжиниринга новые продукты обычно разрабатываются путем повторного использования испытанных деталей, но в программной инженерии кажется, что новые продукты почти всегда разрабатываются с нуля. Чтобы воспользоваться преимуществами возможности повторного использования, необходимо разработать более совершенные методологии, такие как универсальное программирование, которые облегчают эту возможность. Универсальное программирование, применяемое в контексте разработки компьютерного программного обеспечения, представляет собой инновационную парадигму, которая делает программное обеспечение настолько общим, насколько это возможно, без потери эффективности. Благодаря применению этого метода, особенно на этапах проектирования, разрабатывается программное обеспечение, предлагающее высокую степень абстракции, которая применима к широкому спектру ситуаций и областей.

3.1.1 Цели универсального программирования

Применяя универсальное программирование для разработки компьютерного программного обеспечения, важными целями являются:

- повторное использование. Это главная причина для применения идей универсального программирования. Реальная цель, однако, состоит в том, чтобы иметь возможность повторно использовать и широко расширять программные компоненты, чтобы они адаптировались к большому числу взаимосвязанных проблем. Эта концепция повторного

- использования гораздо шире, чем считалось до сих пор;
- качество. Из-за большого потенциала повторного использования универсального программирования необходимо гарантировать максимальное качество. Здесь качество относится к правильности и надежности реализации, которая обеспечивает требуемую степень надежности. Кроме того, универсальное программирование обеспечивает высокую надежность, поскольку реализации представляют собой не более чем скелеты, без деталей, и как таковые проще создавать на этапе кодификации и, следовательно, могут быть созданы с минимальными ошибками;
 - эффективность. Как и в случае с качеством, должна быть гарантирована эффективность компонентов, поскольку, если этого не сделать, будут отмечены последствия для производительности, как и при недостатке качества, во всех задействованных системах;
 - производительность. Неотъемлемой частью повторного использования является экономия за счет отсутствия необходимости заново создавать программные компоненты, которые уже существуют. Следовательно, наблюдается рост производства вычислительной техники, что является одним из преимуществ, к которым стремится универсальное программирование, чтобы реализовать производственный потенциал вычислительных приложений, как и в других отраслях;
 - автоматизация. Здесь цель состоит в том, чтобы автоматизировать процессы таким образом, чтобы общие требования с высоким уровнем абстракции и специально разработанные инструменты могли использоваться для создания оперативных программ. Универсальное программирование обеспечивает основу для первоначальных общих требований. Полученные программы станут основой для других, более специфичных программ, которые, в свою очередь, создадут другие, следуя каскадному процессу, что даст более прочную основу, позволяющую автоматизировать больше процессов;

- персонализация. По мере того, как общие требования становятся более конкретными, создаваемый продукт становится более оптимизированным для удовлетворения конкретных потребностей клиента. Универсальное программирование в значительной степени вмешивается в первую фазу этого цикла абстракции / специализации / персонализации, идентифицируя абстракции и предоставляя первый уровень специализации.

Универсальное программирование также представляет собой один из важных методов создания эффективных продуктовых линеек в соответствии с подходом Product-Line Architecture (PLA). PLA способствует быстрой и дешевой разработке больших семейств связанных программных приложений из компонентов многократного использования. В PLA определенный уровень автоматизации обеспечивается в виде генераторов (также известных как инструменты настройки компонентов) для реализации решений для больших частей разрабатываемых систем.

3.1.2 Пример приложения CSCL с использованием методик универсального программирования

Разработка приложений на основе универсального программирования имеет не только практическое преимущество в виде повышения производительности, но и приводит к созданию более надежного программного обеспечения. Уже были проведены значительные исследования по созданию библиотек универсальных структур данных, полностью основанных на универсальном программировании, но очень мало было сделано для разработки компьютерных систем [17]. Чтобы проверить этот подход, в предыдущих исследованиях рассматривалось создание программной платформы под названием Библиотека общего назначения (БОН). БОН состоит из компонентов широкого использования, что создает основу для построения сложных систем, требующих управления пользователями, взаимодействующими с системой, и оптимизации системных ресурсов. Целью этого исследования было изучение возможности разработки

универсального программного обеспечения в качестве основы сложных и обширных компьютерных систем предметной области. Библиотека была основана на парадигме универсального программирования, поскольку целью было обеспечить максимально возможное повторное использование ее собственных универсальных компонентов для разработки конкретных компьютерных систем высокой сложности. Первым шагом требуется определить те части, которые являются общими для большинства приложений этой области, а затем приступить к выделению фундаментальных частей в форме абстракций, из которых будут получены основные требования. Как только будет произведено логическое разделение на компоненты и подсистемы, каждый компонент будет проанализирован и спроектирован отдельно.

Чтобы сохранить нетронутыми идеи дизайна универсального программирования, был реализован неявный логический уровень, который соответствует идеям универсального программирования.

Библиотека общего назначения состоит из трех компонентов, которые составляют каркас базовой структуры любого приложения, созданного с использованием этой библиотеки.

Компонент «Пользователи». Он состоит из двух подсистем:

- подсистема `GenericUserManagement` управляет основными элементами, участвующими в администрировании пользователей вычислительной системы. Концепция универсального пользователя представляет человека, группу, устройство, систему, организацию и т.д., которые могут иметь несколько удостоверений личности. Каждое удостоверение представляет роль пользователя в определенной настройке, поэтому один и тот же пользователь может иметь несколько ролей. Общий пользователь может также иметь несколько аутентификаторов, которые являются как общедоступными, так и частными, чтобы проверять каждое из удостоверений. Закрытые ключи аутентификаторов будут зашифрованы в целях безопасности, что позволит пользователям

свободно выбирать и внедрять алгоритм шифрования, который наилучшим образом соответствует их конкретным потребностям. Таким образом, универсальный пользовательский объект будет собирать всю доступную информацию о пользователе вычислительной системы, чтобы иметь возможность использовать ее при создании большинства вычислительных приложений, связанных с взаимодействием с пользователем;

- подсистема `UserProfileManagement` предназначена для управления профилями пользователей с информацией о персонализации пользователями настройки (или среды). Под профилем пользователя понимается информация, которая определяет, как конкретный пользователь работает в системе. Эта информация получается через элементы профиля, где каждый элемент задает определенную часть информации о пользователе (псевдоним, язык, размер шрифта и букв, цвета и т.д.), которая моделируется как `ElementProfile`. Сумма всех этих `ElementProfile` образует `UserProfile`. Каждый пользователь может принадлежать к одной или нескольким группам пользователей для обмена информацией (например, команда, выполняющая проект, набор устройств, совместно использующих одни и те же данные и т.д.).

Компонент безопасности. Он состоит из двух подсистем, содержащих полное описание принятых мер и правил, принятых для защиты системных ресурсов от преднамеренного или случайного неправильного использования пользователями и для защиты от несанкционированного доступа к системе:

- подсистема управления аутентификацией предназначена для идентификации пользователя, желающего войти в систему, и управляет процедурами запуска и закрытия пользователей, распознанных системой. Общий процесс аутентификации состоит из запуска сеанса пользователя, в котором, во-первых, создается сеанс неиницированного пользователя, в котором требуются учетные данные пользователя для аутентификации. Пользователи предоставляют свои

аутентификаторы в качестве входных параметров. Исходя из идентификатора, идентифицируется GenericUser, и аутентификатор, хранящийся в системе, сравнивается с тем, который был предоставлен пользователем, и проверяется его действительность;

- подсистема управления авторизацией имеет основной целью администрирование кода безопасности системы. Этот код состоит из всех норм и правил безопасности, установленных в системе, и составляет политику безопасности системы. Политика безопасности означает все нормы и декларации, указанные для определения доступа к ценным ресурсам системы. Эта информация, по сути, будет получена из разрешения, предоставленного пользователям для входа на определенные ресурсы посредством определенных действий. Цель состоит в том, чтобы ограничить доступ пользователей к ценным ресурсам системы. Предоставление разрешения конкретному пользователю повлечет за собой присвоение соответствующего пользователю определенного уровня привилегий на основе существующей информации. Эти привилегии будут назначены во время запуска сеанса, как только пользователь успешно пройдет аутентификацию.

Компонент управления состоит из двух подсистем, содержащих те устройства (например, файл журнала) и процессы (например, вычисление статистики), которые используются при управлении и обслуживании системы для правильного администрирования доступных ресурсов. Он направлен на повышение как производительности, так и безопасности.

- система MonitoringManagementSystem имеет основной целью администрирование всех данных, создаваемых самой системой в результате событий, происходящих во время нормального использования. Ключевым элементом этой подсистемы является история, составленная файловым журналом, который регистрирует все перемещения и инциденты в повседневных операциях системы, таких

как запуск и закрытие пользовательских сеансов, ввод ресурсов, неудачная аутентификация и системные ошибки. Каждое из этих событий записывается в объект ElementLog, содержащий дату и время возникновения события вместе с текстовым описанием. Таким образом, объект истории содержит всю информацию, влияющую на систему, сгенерированную при ее обычном использовании;

- подсистема PerformanceControlManagement имеет основной целью администрирование и поддержание статистических исследований, полученных на основе собственных данных системы о событиях, сгенерированных функционированием системы и сохраненных в истории. Объект статистики предоставит возможность извлекать полезную информацию об этих событиях, которая приведет к определению уровня производительности и надежности критических частей системы. Один объект статистики будет содержать данные истории, которые вместе с объектом критериев будут выступать в качестве источника ссылки для извлечения требуемой информации. Объект критериев может быть множественным, учитывающим различные атрибуты или поля истории, чтобы создавать сложные консультации, которые дадут подробные знания о работе системы. Статистические результаты могут быть предложены в различных форматах в зависимости от потребностей системы и возможностей визуализации, включая текстовую форму, HTML, графики (столбчатые диаграммы, круговые диаграммы и т.д.) и другие доступные форматы. Эти результаты позволят разработать подробные отчеты о состоянии системы, которые укажут путь к внесению соответствующих исправлений в системные отклонения и проведению любых необходимых улучшений.

Пользовательский интерфейс ориентирован в целом таким образом, чтобы сделать возможной детализацию в графическом и текстовом режимах. Несмотря на то, что в этих средах пользовательский интерфейс обычно будет

находиться в графическом режиме, необходимо учитывать эту абстракцию, чтобы сделать логическую часть приложения независимой от конкретного дизайна графического пользовательского интерфейса [30]. Дизайн сохраняемости также является общим, и поэтому абстракция `diskManager` рассматривалась как связующее звено между будущим приложением и его данными, чтобы сделать дизайн сохраняемости независимым от конкретной технологии, которая будет управлять данными и позволит обрабатывать как обычные текстовые файлы, так и базу данных во время детализации. Наконец, полная иерархия исключений обеспечивает высокую степень надежности, не зависящую от обработки ошибок конкретной платформой, поддерживающей программное обеспечение.

Этот подход может быть применен к очень большому числу потенциальных применений, что позволяет извлечь большую выгоду из повторного использования. Действительно, следующие требования чрезвычайно часто встречаются при работе с новым приложением и могут быть выполнены с помощью библиотеки общего назначения:

- управление пользователями, их профилем и рабочими пространствами, а также системными ресурсами;
- механизмы входа /выхода из системы для проверки пользователей. Это позволит защитить систему как от неизвестных пользователей, так и от преднамеренного или случайного неправильного использования ее ресурсов;
- идентификация и уведомление о последних событиях, а также просмотр истории системных событий с целью извлечения статистики о любом периоде жизни системы.

При создании приложения, соответствующего этим требованиям, можно проверить, насколько, даже на этапах проектирования, библиотека общего назначения идеально адаптируется благодаря хорошему соответствию предлагаемых универсальных объектов с экземплярами, описанными здесь. Например, как `GenericUser` совпадает с любым типом пользователя,

найденным в любом приложении (отдельные пользователи, группы и т.д.), а Identity становится уникальным идентификатором пользователя (например, адрес электронной почты, номер социального страхования и т.д.), а также как ElementLog собирает всю информацию, необходимую для идентификации события системы, с помощью которых можно осуществлять их оповещение остальных пользователей. Что касается функциональности, общие процессы также хорошо сочетаются со специфическими процессами, такими как AuthenticationValidation, которые после создания экземпляра позволяют выполнять аутентификацию пользователей и получать доступ к их собственному рабочему пространству и ресурсам. Пользовательский интерфейс создается в графическом режиме, используя преимущества универсальности библиотеки. Что касается сохраняемости, доступна абстракция GenericDiskManager и ее специализации, которые представляют собой мост, который сохраняет логику приложения независимой от его данных, обеспечивая сохраняемость в различных моделях.

3.2 Общий взгляд на приложения для совместного обучения

В этом подразделе дается общее представление о предметной области CSCL путем анализа и учета общности требований большинства сред совместного обучения и, в частности, предоставления эффективных знаний приложениям CSCL.

3.2.1 Общая основа для приложений CSCL

В последние годы произошел взрыв новых приложений CSCL, направленных на создание среды совместного обучения, в которой учащиеся и учителя могут сотрудничать друг с другом для достижения общей цели обучения. Для достижения этой цели, приложения для совместной работы должны обеспечивать поддержку трех основных аспектов: координации, сотрудничества и коммуникации, при этом коммуникация является основой для достижения координации и сотрудничества [7]. Сотрудничество и

коммуникация могут быть синхронными или асинхронными. Первое означает одновременное сотрудничество с типично детализированными уведомлениями, дающими немедленную обратную связь о действиях других участников, в результате чего общий ресурс (такой как текстовый документ и сообщение) не будет иметь срока службы после совместного использования. Последнее означает сотрудничество в разное время, и общий ресурс будет храниться в постоянной поддержке.

Различные области накладываются друг на друга, и любая система совместной работы должна поддерживать все три аспекта:

- координация является важным аспектом любой совместной деятельности. Это влечет за собой сочетание и последовательность независимой в остальном работы для достижения более масштабной цели. В среде совместного обучения координация в основном относится к задачам по формированию учебной группы, а также к определению и планированию групповых целей. Более того, координатор группы может отслеживать статус задачи, крайние сроки, использование ресурсов, результаты работы или другие важные параметры процесса, чтобы правильно руководить группой;
- сотрудничество основывается на том, что учащиеся обмениваются всеми видами документов. Таким образом, совместное использование ресурсов между несколькими участниками является центральной функциональностью систем CSCL. Совместное использование может быть синхронным, когда несколько участников одновременно обращаются к одному и тому же ресурсу (то есть они работают с одной и той же копией документа), или асинхронным, когда разные участники обращаются к одному и тому же ресурсу в разное время (каждый из них работает с другой копией одного и того же документа);
- коммуникация — это еще один функциональный аспект систем совместной работы, направленный на поддержку коммуникации между двумя или более участниками совместного обучения. Общение

включает в себя текстовые сообщения, устные взаимодействия или невербальные обмены, такие как жесты во время видеоконференции. Общение может происходить асинхронно (разные участники общаются в разное время, например, по электронной почте, в дебатах и т.д.) или синхронно (участники общаются одновременно, например, в чате, видеоконференции и т.д.). Поддержка связи основана на задействованных четырех элементах: сообщение как носитель информации между процессом-отправителем и процессом-получателем (который получает и, возможно, обрабатывает сообщение) по каналу. Более того, в этом контексте необходимо реализовать различные способы адресации сообщений, такие как двухточечный, многоадресный и широковещательный.

- осведомленность необходима для любой из трех форм сотрудничества, рассмотренных выше. Это обеспечивает неявную координацию совместного обучения, возможности для неформального, спонтанного общения и дает пользователям необходимую обратную связь о том, что происходит в системе. В частности, с одной стороны, синхронная осведомленность позволяет пользователям точно знать, что делают другие соучастники (например, во время сеанса совместного редактирования видно, кто что редактирует) и когда документы используются другими. С другой стороны, асинхронная осведомленность определяет, кто, когда, как и где создавал, изменял или читал общие ресурсы другими.

Чтобы улучшить сотрудничество внутри группы, важно учитывать, как текущее, так и будущее поведение всех типов пользователей, а также тот факт, что цели или намерения пользователей могут меняться по мере их взаимодействия с системой [19]. С этой целью важно разработать своего рода пользовательские и групповые модели, описывающие, например, характеристики пользователя, намерения, убеждения, знания, навыки, роли и совместную деятельность. Более того, пользовательские и групповые модели

должны быть достаточно открытыми, чтобы позволять добавлять к ним новые сервисы и совместные действия в соответствии с потребностями участников [31].

Дизайн пользовательского интерфейса CSCL создает гораздо больше проблем, чем дизайн интерфейсов для однопользовательских приложений (например, многопользовательских редакторов). Пользовательский интерфейс должен предоставлять информацию о том, что делают другие участники, для эффективной поддержки совместных задач, и должна быть представлена дополнительная информация [25]. Последнее относится к последствиям действий других пользователей, о которых необходимо сообщать с помощью визуальных или звуковых сигналов. Таким образом, пользовательский интерфейс является основным способом поддержки осведомленности в многопользовательских средах совместной работы.

Хотя большинство исследовательских усилий в областях CSCL было посвящено разработке условий дистанционного обучения, большинство учебных мероприятий по-прежнему проводятся в традиционном очном классе. С этой целью общий подход, подобный подходу этого тезиса, должен поддерживать общую основу для обоих сценариев, и возможно создавать приложения CSCL как для виртуального обучения (т.е. большинство участников физически находятся в разных местах), так и для традиционного обучения (т.е. все участники физически находятся в одном и том же месте, обычно в классе). В этом тезисе, хотя он в основном относится к виртуальным средам CSCL, принципы одинаковы для обоих сценариев.

3.2.2 Важность анализа взаимодействия в практике CSCL

Среды совместного обучения характеризуются высокой степенью взаимодействия пользователя и системы, что порождает огромное количество событий действия. Управление событиями действий является ключевой проблемой в приложениях, поскольку, с одной стороны, анализ данных, собранных из реальных ситуаций совместного обучения в режиме онлайн, помог бы понять важные вопросы функционирования группы и процесса

совместного обучения. Кроме того, это может помочь как при проектировании более функциональных рабочих пространств и программных компонентов, так и при разработке более совершенных средств, таких как осведомленность, обратная связь, мониторинг рабочего пространства, оценка и отслеживание работы группы координатором, наставником и т.д. Действительно, путем фильтрации данных адекватное управление событиями позволяет создать список параметров, которые могут быть использованы для анализа действий в групповом пространстве (например, поток общения между преподавателем и группой или между членами группы, асинхронность в групповом пространстве и т.д.). Эти параметры позволили бы повысить эффективность групповой деятельности и спрогнозировать поведение группы и индивидуальные установки ее членов в общем рабочем пространстве.

Кроме того, при разработке приложений необходимо правильно организовать и администрировать как ресурсы, предлагаемые системой, так и пользователей, получающих доступ к этим ресурсам. Все это взаимодействие пользователя с ресурсом и пользователя с пользователем генерирует события или журналы, которые собираются в лог-файлах и представляют собой информационную основу для выполнения статистических процессов, направленных на получение полезных знаний о системе. Это облегчит процесс совместного обучения, информируя пользователей о том, что происходит в системе (например, вклад других людей, созданные новые документы и т.д.) и контроль поведения пользователей с целью оказания им поддержки (например, помощь учащимся, которые не в состоянии выполнить задачу самостоятельно). Таким образом, взаимодействие между пользователем и пользовательскими ресурсами имеет решающее значение в любой учебной среде для совместной работы, чтобы группы учащихся могли общаться друг с другом и достигать общих целей (например, совместная деятельность в классе).

Хотя взаимодействие с пользователем является наиболее важным моментом для управления в приложениях, обычно также важно иметь

возможность отслеживать и контролировать производительность и общее функционирование системы. Это позволит администратору постоянно отслеживать критические части системы и действовать в случае необходимости. Кроме того, это добавляет неявный уровень безопасности к тому, который уже существует (например, контроль привычек пользователей, позволяющий обнаруживать мошенническое использование системы неавторизованными пользователями).

Чтобы эффективно передавать знания, полученные в результате групповой деятельности, пользователям с точки зрения осведомленности и обратной связи, приложения CSCL должны обеспечивать полную поддержку вышеупомянутых трех основных аспектов, существующих в любом совместном приложении, а именно координации, коммуникации и совместной работы, чтобы создавать виртуальные среды где учащиеся и учителя могут сотрудничать друг с другом для достижения общей цели обучения. Координация включает в себя организацию группы для достижения поставленных целей и мониторинг активности пользователей, что возможно за счет поддержания осведомленности участников. Коммуникация связана с поддержкой связи в основном посредством сообщений между пользователями внутри групп и между ними и может осуществляться как в синхронном, так и в асинхронном режимах. Наконец, совместная работа позволяет членам группы совместно использовать любые ресурсы, причем как в синхронном, так и в асинхронном режимах. Как координация и сотрудничество, так и коммуникация приведут к возникновению множества событий, которые будут доведены до сведения пользователей после того, как эти события будут обработаны и проанализированы, чтобы предоставить пользователям как можно больше немедленной информации и постоянный поток как можно большего количества отзывов.

3.3 Вычислительная модель для анализа данных и управления ими в Библиотеке совместного обучения

Основным вкладом этого тезиса является универсальная, многогранная, надежная, гибкая, совместимая, основанная на компонентах и ориентированная на обслуживание платформа под названием «Библиотека совместного обучения» (БСО). БСО основан на парадигме общего программирования, позволяющей полностью и эффективно повторно использовать его общие компоненты в качестве основы для построения любого приложения для совместного обучения. Эта универсальная платформа реализует концептуализацию фундаментальных потребностей, существующих в любом опыте совместного обучения. В контексте этого исследования эта платформа особенно используется в качестве вычислительной модели для встраивания информации и знаний из приложений CSCCL групповой деятельности в качестве реализации концептуальной модели для анализа данных и управления.

По этой причине при разработке БСО особое внимание уделяется анализу событий и управлению ими. С этой целью разработан специальный компонент, позволяющий собирать, структурировать и классифицировать всю сгенерированную информацию о событии, а также обрабатывать эту информацию для ее последующего анализа. Этот компонент полностью определяет и реализует первые два этапа вышеупомянутого процесса внедрения информации и знаний. Третий этап, представление (для информирования пользователей и обратной связи), полностью выполняется другим компонентом, который реализует три основных элемента, задействованных в любом приложении группового программного обеспечения (т.е. координация, коммуникация и сотрудничество), и отвечает за передачу полученных знаний пользователям с точки зрения осведомленности и обратной связи.

Чтобы соответствовать этим требованиям, разработка БСО основана на

парадигме Model-Driven Development (MDD) и поддерживающей ее платформе, а именно Model-Driven Architecture (MDA). Предлагая MDA, при разработке БСО планируется использовать две ключевые идеи, которые оказали значительное влияние на решение текущих задач в области разработки программного обеспечения: сервис-ориентированные архитектуры (SOA) и архитектуры продуктовой линейки (PLA). Что касается первого, SOA обеспечивает большую гибкость системным архитектурам, организуя систему как набор инкапсулированных сервисов. Следовательно, SOA полагается на сервисы, предоставляемые компонентом, которые будут использоваться любыми другими компонентами на основе только контракта интерфейса. Что касается последнего, PLA способствует быстрой и дешевой разработке больших семейств связанных программных приложений из повторно используемых компонентов. В PLA определенный уровень автоматизации обеспечивается в виде генераторов (также известных как инструменты настройки компонентов) для реализации решений для больших частей разрабатываемых систем. Принимая во внимание эти подходы, БСО основан на SOA и парадигме общего программирования как центральной части разработки в MDD.

В частности, при разработке БСО сначала будет создана платформенно-независимая модель (PIM) с применением следующих общих идей программирования:

- определить семантику свойств и понятий предметной области;
- извлечение и обозначение общих и переменных свойств и их зависимостей в виде абстракций, найденных в домене CSCL;
- выделение фундаментальных частей в виде абстракций, из которых будут получены, проанализированы и спроектированы основные требования в виде традиционной трехуровневой архитектуры (т.е. презентация, бизнес и информация).

С этой целью, PIM был выражен с использованием UML в качестве стандартного языка моделирования (рисунок 3).

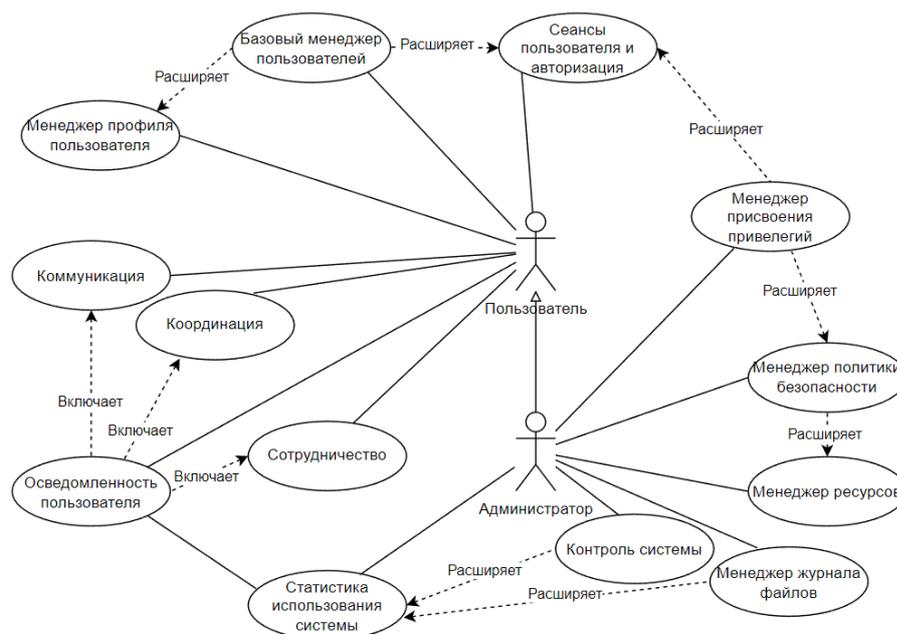


Рисунок 3 - Диаграмма вариантов использования на основе UML с общими требованиями БСО

Конечная цель БСО - обеспечить полное и эффективное повторное использование его общих сервисов и компонентов в качестве основы для создания любого приложения для совместного обучения, и, в частности, приложений CSCL. Таким образом, эта платформа реализует концептуализацию фундаментальных потребностей, существующих в любом опыте совместного обучения. Кроме того, БСО обладает высокой степенью совместимости в распределенных средах, обеспечивая полную гибкость предлагаемых услуг с точки зрения языков реализации и базовых программных и аппаратных платформ.

В остальной части этого раздела подробно описывается модель РИМ на основе UML для БСО посредством общего представления архитектуры БСО.

3.3.1 Архитектура Библиотеки совместного обучения

БСО состоит из пяти компонентов, отвечающих за управление пользователями, администрирование, безопасность, управление знаниями и функциональность, которые отображают основные проблемы, связанные с любым приложением для совместного обучения.

- Компонент управления пользователями CSCL: содержит функционал, связанный с управлением пользователями в приложениях, которые могут выступать в качестве координатора группы, члена группы, групповой сущности и системного администратора. Он будет выполнять как основные функции управления пользователями в среде обучения (а именно регистрацию, снятие с регистрации, модификации, присоединение к группе или встречи с членами группы), так и управление профилем пользователя.
- Компонент управления безопасностью CSCL: содержит все общие описания мер и правил, принятых для решения проблем аутентификации и авторизации и, таким образом, защиты системы как от неизвестных пользователей, так и от преднамеренного или случайного неправильного использования ее ресурсов.
- Компонент управления администрированием CSCL: содержит конкретные данные из файлов журналов и те анализы (т.е. статистические вычисления), которые необходимы для выполнения всего системного контроля и обслуживания для правильного администрирования системы и улучшения ее с точки зрения производительности и безопасности. Более того, он будет управлять ресурсами совместной рабочей области, которыми может управлять член группы, действующий в качестве администратора внутри группы.
- Компонент управления знаниями CSCL: он управляет всеми конкретными и крупными пользовательскими событиями, чтобы обрабатывать данные взаимодействия с пользователем как важную информацию для извлечения необходимых знаний, чтобы уведомлять пользователей о том, что происходит в системе, а также отслеживать поведение пользователя и управлять системными ресурсами. С этой целью этот компонент был разделен на подсистемы управления деятельностью CSCL и обработки знаний CSCL. Первый направлен на сбор и классификацию пользовательских событий, зафиксированных в

соответствии с полной иерархией пользовательских событий, которые основаны на вышеупомянутых трех общих параметрах групповой деятельности: выполнение задач, групповое функционирование (т.е. поведение при взаимодействии) и построение шаблонов. Последний отвечает за выполнение статистического анализа ранее обработанной информации о событии и включает в себя другую общую иерархию, которая содержит те статистические критерии, которые наиболее распространены в этих средах (например, количество студентов, подключенных за определенный период времени, средняя рабочая сессия студента). Кроме того, это позволит экспортировать и извлекать информацию журнала в различных форматах для последующего статистического анализа во внешних статистических пакетах. Конечная цель этого компонента - извлечь ценную информацию из событий, сгенерированных с целью раскрытия полезных знаний. Поскольку этот компонент представляет собой ядро БСО, он будет очень подробно объяснен позже в следующих разделах.

- Функциональный компонент CSCL: вместе с предыдущим компонентом он формирует основу среды совместного обучения, определяя три основных элемента, задействованных в любом групповом программном обеспечении, а именно координацию, коммуникацию и сотрудничество. Данные области, зачастую могут пересекаться друг с другом, и потому любая система совместной работы должна поддерживать все три аспекта. Из-за их важности этот компонент предоставляет несколько подсистем или модулей для обеспечения прямой поддержки каждой из этих областей, а именно координации CSCL, коммуникации CSCL и сотрудничества CSCL. Модуль поддержки координации предлагает основные инструменты для облегчения групповой организации в планировании и достижении целей участников, а также группового мониторинга путем моделирования осведомленности ее участников. Модуль поддержки связи включает в себя четыре основных элемента:

отправителя, сообщение, канал и получателя, и может быть реализован несколькими способами в зависимости от средств передачи сообщений (точка-точка, многоадресная рассылка и широковещательная передача). Более того, каждое сообщение может быть доставлено асинхронно (как в случае электронной почты, где сообщение по умолчанию становится постоянным) или синхронно (как в чате, где разговор становится постоянным, чтобы его можно было позже обработать). Наконец, модуль поддержки совместной работы позволит участникам совместно использовать программные и аппаратные ресурсы как в синхронном (например, редакторы реального времени), так и в асинхронном (например, общий доступ к файлам) режимах. Этот компонент также поддерживает представление информации, управляемой компонентом управления знаниями CSCL, посредством подсистемы, называемой осведомленностью CSCL, с целью предоставления участникам немедленной осведомленности о том, что происходит в группе. Кроме того, в последние несколько лет обратной связи уделяется много внимания из-за ее положительного влияния на совместное обучение в режиме онлайн в таких областях, как групповая мотивация, взаимодействие или способность решать проблемы. Эта характеристика также поддерживается в этом компоненте другой подсистемой, называемой обратной связью CSCL, которая также использует знания, извлеченные из групповой деятельности, чтобы обеспечить участникам постоянный поток максимально возможной обратной связи.

Эти компоненты БСО могут повторно использоваться при создании конкретных, эффективных, надежных, мультиплатформенных и многоазовых сред CSCL, в особенности, веб-приложений. Эти компоненты, в свою очередь, повторно используют более универсальную библиотеку общего назначения (БОН), требования к которой заключаются в высокой степени взаимодействия пользователей и системы, а также оптимального управления системными ресурсами. Включив область совместного обучения в

БОН, БСО будет представлять собой конкретизацию БОН (рисунок 4).

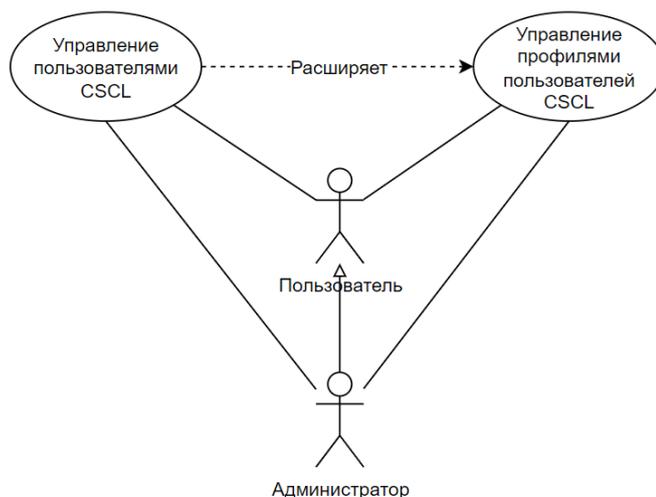


Рисунок 4 - Повторное использование на уровне требований как часть модели РІМ БОН

БОН предоставляет БСО следующее и важное поведение по умолчанию:

- чтобы улучшить сотрудничество внутри группы, важно учитывать, как текущее, так и будущее поведение всех типов пользователей, а также возможные изменения целей и намерений пользователей по мере их взаимодействия с системой. С этой целью были разработаны общие модели пользователей и групп для описания характеристик пользователей, намерений, убеждений, знаний, навыков, ролей и совместной деятельности, среди прочего. Более того, пользовательские и групповые модели достаточно открыты, чтобы позволять добавлять новые услуги и совместные действия в соответствии с потребностями участников;
- дизайн пользовательского интерфейса в приложениях для совместного обучения требует большей проработки, чем дизайн интерфейсов для однопользовательских приложений. Интерфейс должен предоставлять информацию о том, что делают другие участники, чтобы эффективно поддерживать совместные задачи, а информация о действиях других

пользователей должна передаваться с помощью визуальных или звуковых сигналов. Кроме того, пользовательский интерфейс должен быть ориентирован в целом таким образом, чтобы сделать возможной детализацию в графическом и текстовом режимах. Несмотря на то, что пользовательский интерфейс в веб-средах совместного обучения обычно выполняется в графическом режиме, необходимо учитывать общую направленность, чтобы сделать логическую часть приложения независимой от конкретного дизайна графического пользовательского интерфейса;

- дизайн сохранения в БСО также является общим, и поэтому была рассмотрена абстракция диспетчера дисков. Диспетчер дисков действует как связующее звено между будущим приложением и его данными, чтобы сделать дизайн хранилища независимым от конкретной технологии, которая будет управлять данными. Таким образом, при детализации можно обрабатывать как обычные текстовые файлы, так и различные системные менеджеры баз данных. Кроме того, полная технологически независимая концептуальная модель данных предоставляется как часть РИМ, которая может быть реализована в различных технологиях, управляющих общей сохраняемостью;
- надежность обеспечивается за счет полной иерархии обработки ошибок, и поэтому гарантируется высокая степень качества и надежности компонентов без зависимости от обработки ошибок конкретной платформы, поддерживающей программное обеспечение.

Вся деятельность CSCL выполняется в пяти описанных компонентах, которые представляют собой вычислительную модель, которая реализует концептуализацию фундаментальных потребностей, существующих в приложениях, особенно для полного выполнения процесса, описанного в предыдущих работах, о внедрении информации и знаний о групповой деятельности в приложения CSCL эффективным и действенным образом. Действительно, эти области характеризуются высокой степенью

взаимодействия с пользователем. Это взаимодействие генерирует множество событий, которые будут обрабатываться и анализироваться компонентом управления знаниями CSCL, чтобы извлекать полезные знания из групповой деятельности, а затем динамически уведомлять участников об этих знаниях о действиях других посредством функционального компонента CSCL. Это полностью выполняется двумя компонентами, связанными с управлением знаниями и поддержкой функциональности, а именно управлением знаниями CSCL и функциональностью CSCL.

3.3.2 О преимуществах использования сервис-ориентированных архитектур для CSCL

Вторая PSM БСО была разработана в соответствии с принципами сервис-ориентированной архитектуры (SOA) и реализована с использованием веб-сервисов.

SOA представляет собой следующий шаг в разработке программного обеспечения, который поможет организациям удовлетворить их постоянно усложняющийся набор потребностей и задач, особенно в распределенных системах. Это достигается путем динамического отключения и вызова соответствующих служб для выполнения запроса из разнородных сред, независимо от деталей и различий этих сред. Делая сервис независимым от контекста, SOA предоставляет программному обеспечению важные нефункциональные возможности для распределенных сред (такие как масштабируемость, гетерогенность и открытость) и значительно упрощает процессы интеграции.

SOA полагается на сервисы. Сервисы представляют собой модель поведения, которая предоставляется поставщиком и используется любыми пользователями, основанное только на контракте интерфейса. В рамках SOA сервисы:

- независимы от местоположения, позволяя внедрять, тиражировать и перемещать службы на другие компьютеры без ведома пользователя;
- включают в себя динамический доступ, поскольку службы

расположены, привязаны и вызываются во время выполнения;

- содействуют интероперабельности, позволяющей различным организациям, поддерживаемым разнородными аппаратными и программными платформами, совместно использовать одни и те же сервисы;
- облегчают интеграцию других существующих систем;
- полагаются на инкапсуляцию, поскольку они независимы от других сервисов и их содержимого;
- повышают гибкость за счет возможности замены служб без ущерба для задействованных базовых систем;
- позволяют создавать новый сервис из других микро-сервисов.

Организации, использующие ключевые свойства SOA, получают множество преимуществ. Благодаря независимости от местоположения, а также динамическому обнаружению и вызову сервиса, мобильность программного обеспечения становится реальностью. Это позволяет организациям гибко перемещать службы на разные компьютеры, не оказывая влияния на базовую систему. Кроме того, независимости от местоположения также способствует масштабируемости и доступности без ведома клиента, поскольку можно как масштабировать количество экземпляров нескольких служб, запущенных на нескольких серверах, так и поддерживать отказоустойчивость, перенаправляя запрос на реплику, когда сервер недоступен.

Несмотря на то, что SOA может быть реализована с помощью других технологий, за последние несколько лет веб-сервисы стали играть главную роль в SOA, предоставляя набор стандартных протоколов, которые отвечают основным потребностям SOA, таких как XML, UDDI, WSDL, SOAP и HTTP. Это связано с тем, что эти протоколы позволяют сервису быть независимым от платформы и языка, динамически размещаться и вызываться, а также взаимодействовать в сетях различных организаций. В результате технология веб-сервисов обеспечивает более низкие затраты на интеграцию наряду с

гибкостью и упрощением настройки.

Существует много общего между потребностями в совместном обучении и преимуществами, предоставляемыми SOA. В результате такого сопоставления SOA представляется наилучшим выбором для поддержки разработки наиболее распространенных и сложных сред совместного обучения. В контексте CSCL SOA расширяет возможности образовательных организаций за счет повышения гибкости их педагогических стратегий, которые могут быть постоянно адаптированы, скорректированы и персонализированы для каждой конкретной целевой учебной группы. Более того, SOA облегчает повторное использование успешного опыта совместного обучения и позволяет участникам совместного обучения легко адаптировать и интегрировать свои текущие лучшие практики и существующие хорошо известные инструменты обучения в новые цели обучения.

Поэтому, чтобы повысить гибкость и интероперабельность, вторая PSM БСО опирается на SOA, поскольку она представляет собой идеальный контекст для поддержки и использования как последних тенденций разработки программного обеспечения, так и преимуществ, предоставляемых распределенными системами, для полного удовлетворения жестких требований приложений CSCL. Использование SOA в контексте БСО дает следующие ключевые преимущества:

- упрощает механизм инкапсуляции, который необходим для общего взаимодействия различных реализаций;
- адаптирует приложения CSCL к меняющимся технологиям;
- легко интегрирует приложения CSCL с устаревшими системами и инструментами обучения;
- обновляет педагогические модели и инструменты обучения, не оказывая влияния на базовые системы и платформы обучения;
- быстро и легко создавать и обновлять учебный процесс на основе существующих сервисов.

Веб-сервисы были технологией реализации, выбранной для PSM этого

БСО, учитывая широко принятые протоколы и стандарты, что представляет собой обоснование подхода к веб-сервисам. Эти стандарты представляют собой подходящий контекст для обеспечения интероперабельности и масштабируемости за счет использования больших преимуществ распределенных технологий. В результате получается набор файлов WSDL, организованных в каталоги, которые автоматически превращаются в универсальные функциональные веб-сервисы, реализованные на желаемом языке программирования и позволяющие разработчикам реализовывать эти сервисы в соответствии с конкретными требованиями.

Подводя итог, можно сказать, что сочетание MDA, SOA и веб-сервисов приводит к созданию модели, специфичной для платформы (PSM), представляющей собой набор файлов WSDL, организованных в каталоги.

Конечная цель БСО - обеспечить полное и эффективное повторное использование его общих сервисов и компонентов в качестве основы для создания любого приложения для совместного обучения, и, в частности, приложений CSCL. Таким образом, эта платформа реализует концептуализацию фундаментальных потребностей, существующих в любом опыте совместного обучения. Кроме того, БСО обладает высокой степенью совместимости в распределенных средах, обеспечивая полную гибкость предлагаемых услуг с точки зрения языков реализации и базовых программных и аппаратных платформ.

3.4 Веб-сервисы и grid-технологии для улучшения CSCL

Хотя SOA может быть реализовано с помощью других технологий, за последние несколько лет веб-сервисы стали играть важную роль в SOA благодаря более низким затратам на интеграцию наряду с гибкостью и упрощением конфигурации.

Базовая структура веб-сервисов формируется набором широко принятых протоколов и стандартов, которые обеспечивают подходящую

технологии для реализации ключевых требований SOA. Это происходит потому, что используемые протоколы позволяют сервису быть независимым от платформы и языка, динамически размещаться и вызываться, взаимодействовать в сетях различных организаций и поддерживаться крупными организациями.

Кроме того, при обслуживании множества запросов от сильно распределенного сообщества возникает проблема организации и управления многочисленными распределенными аппаратными и программными компонентами. По этой причине введен термин сервис-ориентированная инфраструктура для обозначения механизмов управления ресурсами и предоставления ресурсов, используемых для достижения целей в области качества обслуживания компонентов и приложений. Сетевые сервисы предназначены для достижения этой цели.

Grid-сервисы - это, по сути, веб-сервисы со специфическими расширениями или интерфейсами для использования в grid-системах. Grid-сервисы играют центральную роль в Open Grid Services Infrastructure (OGSI), которая призвана обеспечить инфраструктурный уровень для Open Grid Services Architecture (OGSA). По сути OGSI, grid-сервис - это веб-служба, которая соответствует набору соглашений для таких целей, как управление сроком службы сервиса, проверка и уведомление об изменениях состояния сервиса. Grid-сервисы обеспечивают контролируемое управление распределенным и долговечным состоянием, которое обычно требуется в распределенных приложениях.

Рассмотрим наиболее распространенные существующие потребности в электронном обучении, выявленные в образовательных организациях. С этой целью приводится обзор основных сервисов, необходимых для поддержки приложений для совместного обучения. Представленные услуги были выбраны путем сопоставления наиболее успешных платформ и систем электронного обучения, затем были опущены сервисы, не относящиеся конкретно к CSCL, наконец, были добавлены специфичные для CSCL

сервисы. Разработанное решение (а именно grid-архитектура) отображено на рисунке 5.

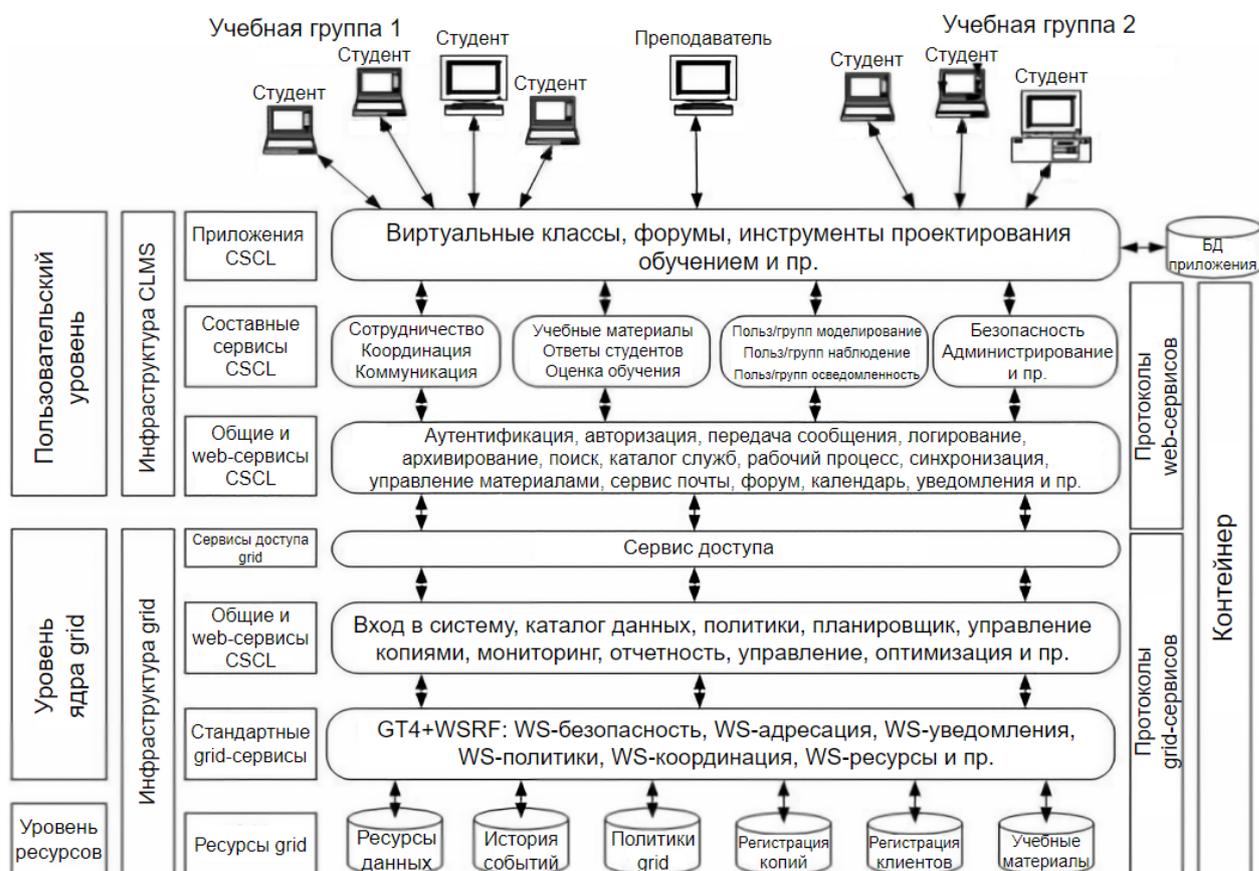


Рисунок 5 – Grid-архитектура для CLMS

С момента замены OGSИ в пользу Web Service Resource Framework (WSRF) grid-сервисы были заменены веб-сервисами с отслеживанием состояния (среди прочих функций), и термин grid-сервис устарел. WSRF регулирует взаимосвязь между веб-службами и ресурсами с отслеживанием состояния с помощью набора соглашений, выраженных в спецификациях составных веб-служб, таких как стандарт WS-Addressing. Однако, термин grid-сервис все еще используется для того, чтобы проводить различие между теми сервисами, которые задействованы на базовом уровне grid, и типичными веб-сервисами без сохранения состояния, реализующими grid-приложения на уровне пользователя.

Чтобы предоставить читаемый и полезный набор сервисов, они сгруппированы вместе в соответствии с теми же критериями, что и сами фреймворки. Таким образом, в этом пункте основное внимание уделяется двум основным уровням обслуживания, а именно общим службам и службам приложений.

3.4.1 Общие grid-сервисы

Сервисы, описанные в этом подпункте, являются универсальными, так что они могут стать основой для любой среды электронного обучения и могут быть общими для нескольких областей применения. Общие службы предоставляют функциональность более низкого уровня, которая не является специфичной для образования, но от которой зависят сервисы и пользователи образовательной области:

- аутентификация. Получение необходимых учетные данные от пользователя, подтверждение их подлинности и выдача разрешения на вход пользователя в систему;
- авторизация. Разрешение приложению устанавливать и запрашивать привилегии пользователя для просмотра, создания или изменения данных приложения или использования функциональных возможностей приложения;
- передача сообщений. Разрешение широковещательной рассылки сообщений пользователям и группам с использованием соответствующей коммуникационной технологии. При этом не требуется заранее принимать конкретный механизм доставки, который будет использоваться при реализации сервиса;
- логирование. Включение отслеживания любой другой службы, и регистрация соответствующей информации и событий по всей системе, с целей диагностики, повышения производительности и информирования пользователей;
- идентификатор. Отвечает за создание и предоставление доступных идентификаторов объектов обучения;

- архивирование. Поддержка доступа к удаленным хранилищам для хранения и извлечения произвольного статического контента;
- рабочий процесс. Обеспечение управления взаимозависимой последовательностью действий, каждое из которых имеет ограничения по завершению;
- поиск. Позволяет находить учебные материалы и другую связанную с ними информацию, получаемую из системы;
- каталог служб. Хранит информацию о таких объектах, как службы, другие хранилища, люди и организации, и оказывает поддержку в поиске доступных служб;
- агент. Это абстракция агента, устройства и т.д., которая может включать в себя базовую информацию, такую как идентификатор, имя, тип, роль, свойства и контактную информацию.
- пользовательские настройки. Предоставляет информацию о личных настройках интерфейса пользователя и позволяет пользовательским агентам, автоматически настраивать себя для конкретных конечных пользователей и избавлять конечных пользователей от необходимости вводить свои настройки в несколько пользовательских агентов;

3.4.2 Прикладные grid-сервисы

Сервисы этой категории зависят от образовательного домена и обеспечивают функциональность, требуемую агентами. Прикладные сервисы могут быть реализованы таким образом, чтобы у них был какой-то пользовательский интерфейс. Их ключевым требованием является предоставление их функциональности для повторного использования любым количеством агентов или других прикладных служб при реализации стандартного интерфейса для поддержки такого повторного использования:

- синхронизация. Определение структуры данных и интерфейсов, ответственных за описание набора возможных последовательностей представления для коллекции ресурсов контента;
- управление материалом. Обеспечение механизмами для создания,

- управления и публикации материалов;
- оценивание. Обеспечение записи оценки, комментирования, посещаемости и баллов для учащегося или группы;
 - группа. Управление созданием, удалением, обновлением и чтением групп;
 - участник. Управление созданием, удалением, обновлением и чтением участников группы;
 - управление курсом. Управление созданием, чтением, обновлением и удалением единиц обучения, курсов, модулей, а также управление информацией о людях, членстве в единицах обучения и пр.;
 - сотрудничество. Абстрактный сервис, поддерживающий конкретную синхронную и асинхронную работу сервисов, таких как форум и чат;
 - координация. Абстрактный сервис, поддерживающий формирование конкретной учебной группы, а также определение и планирование целей группы;
 - коммуникация. Абстрактный сервис, поддерживающий конкретное взаимодействие между пользователями, например, сервис электронной почты;
 - уведомление. Абстрактный сервис, сообщающий пользователям /группам о том, что происходит в процессе обучения, например, службы оповещения и присутствия.

3.4.3 Основные сервисы сетевой инфраструктуры

Существует множество сервис-ориентированных инфраструктур для использования в сетях. Объединив их с CSCCL, можно создать основные grid-сервисы для совместного обучения:

- вход в систему. Обеспечивает аутентификацию, авторизацию и контроль доступа к данным и вычислительным ресурсам;
- каталог данных. Реализует поиск наборов данных на основе метаданных;
- поиск. Представляют собой основную точку входа для доступа к grid.

Позволяет осуществлять динамический поиск сетевых служб, устраняя необходимость заранее знать их местоположение;

- политики. Установка прав доступа, правил и разрешений, позволяющих пользователям, агентам и приложениям использовать службы grid.
- планировщик. Определяет план выполнения задания для grid-приложений. Также отправляет план в grid, для выполнения;
- управление копиями. Повышает надежность и масштабируемость grid, предоставляя возможность копировать и перемещать данные по grid;
- управление и оптимизация. Позволяет запросам на выполнение заданий адаптироваться к динамической среде сети.
- мониторинг. Предоставление информации о текущем состоянии плана работы, отслеживание текущего состояния задачи и затрачиваемых ресурсов.
- отчетность. Обеспечение доступа к ресурсам в соответствии с политиками.

Цель предложенных решений состоит в том, чтобы стать средством эффективного управления и анализа данных взаимодействия для создания универсального механизма разработки программного обеспечения на основе повторного использования компонентов приложений CSCL

Таким образом, в данном разделе исследования были решены следующие задачи:

- разработана концептуальная модель вычислительной системы в форме общей спецификации;
- смоделирована структура эффективного управления и анализа данных взаимодействия, генерируемых в ходе практики CSCL;
- усовершенствован метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL для полуавтоматического генерирования реализаций на основе независимых спецификаций.

4 Объединение CSCL, универсального программирования и распределенных вычислений: оценка и результаты

Ввиду сложности реализации системы, а также соответствующей длительной подготовки перед введением системы в эксплуатацию, и длительным сбором информации, эффективность и испытания разработанного решения будут проходить только для grid-решения данной диссертации. Данные, которые использовались для проведения испытаний, были взяты из открытых источников университетов, в которых уже применяются CSCL-решения.

4.1 Проведение эксперимента и результаты распараллеливания обработки файлов журнала

Для проведения сравнительного исследования между последовательным и grid-подходами была разработана специальная тестовая батарея, в которой использовались как большие объемы информации о событиях, так и хорошо стратифицированные короткие выборки. В этом подразделе представлены экспериментальные результаты, достигнутые с помощью прототипов Grid, в то время как в следующем разделе анализируются некоторые важные аспекты этих результатов, которые необходимо рассмотреть.

Чтобы протестировать эффективность гипотетического решения и сравнить результаты с последовательным подходом, были запущены тестовые лог-файлы из других CSCL-приложений (взятых из открытых источников) на платформе grid в узлах PlanetLab (что представляет собой группу компьютеров, доступных в качестве испытательного стенда для исследования компьютерных сетей и распределенных систем). С этой целью, как упоминалось ранее, были использованы существующие файлы журналов, составляющие групповую деятельность. Другие тесты включали в себя несколько файлов журнала с определенным размером файла журнала и

сложностью события, образующих выборку для каждой репрезентативной стратегии.

Линейность, обнаруженная во времени обработки при последовательном подходе, значительно упростила эксперимент за счет использования одного и того же файла журнала событий в качестве входных данных для всех тестов grid-системы в эксперименте. Затем параметры, касающиеся как количества рабочих станций, так и размера задач (выраженных в количестве событий), были оставлены для изменения. Затем эти параметры должны были быть выполнены рабочими станциями. Тесты были проведены для разного количества рабочих станций с разными размерами задач.

Параллельное ускорение используется для измерения прироста производительности от распараллеленного выполнения по сравнению с его последовательным выполнением и определяется как (1).

$$S(s, p) = \frac{T_s(s)}{T_p(s, p)}, \quad (1)$$

где $S(s, p)$ – параллельное ускорение;

s – размер файла журнала;

$T_s(s)$ – общее время последовательного выполнения для файла журнала размера s ;

$T_p(s, p)$ – общее время параллельного выполнения для файла журнала размера s с использованием p процессоров.

Для тестирования прототипа grid, были использованы лог-файлы, имеющие относительно небольшой размер и относительно низкую частоту сложных событий, но с высокой вариативностью размера файла. С этой целью на PlanetLab устанавливается сервисный контейнер Grid GT3. Более того, рабочий был реализован как простой grid-сервис, который был развернут в контейнере GT3. Затем был взят простой Java-клиент, который играет роль мастера, отправляя задачи простым вызовом операций, предоставляемых

службами рабочих grid-системы следующим образом:

- сетевая служба «рабочий» публикует интерфейс только с одной операцией, которую мастер вызывает для отправки задачи рабочему. Эта операция, которая передает в качестве входных данных текстовое представление событий, подлежащих обработке этой задачей, и возвращает структуру данных, содержащую информацию о производительности выполненной задачи (т.е. прошедшее время, количество обработанных событий и количество обработанных байт);
- мастер – является простым Java-приложением, которое считывает из файла конфигурации папку, содержащую файлы журнала событий для обработки, количество доступных рабочих станций, количество используемых рабочих станций и размер задачи, которая будет отправлена каждому из них. Затем мастер поступает следующим образом: он выбирает из файла конфигурации столько рабочих, сколько необходимо, и помещает их всех в очередь незанятых рабочих. Затем он входит в цикл, считывающий события из файлов журнала событий, и каждый раз, когда он считывает несколько событий, он либо ожидает работника, если очередь пуста, либо вызывает рабочую операцию. Как только вызов рабочему возвращается, рабочий помещается обратно в очередь незанятых работников. Мастер выходит из цикла, когда все события в файлах журнала событий прочитаны и все задачи, которые были отправлены, завершены.

На рисунке 6 показано максимальное ускорение, достигнутое для наблюдаемой полосы пропускания между главным процессором и узлами PlanetLab на момент проведения эксперимента и для различного количества протестированных рабочих.

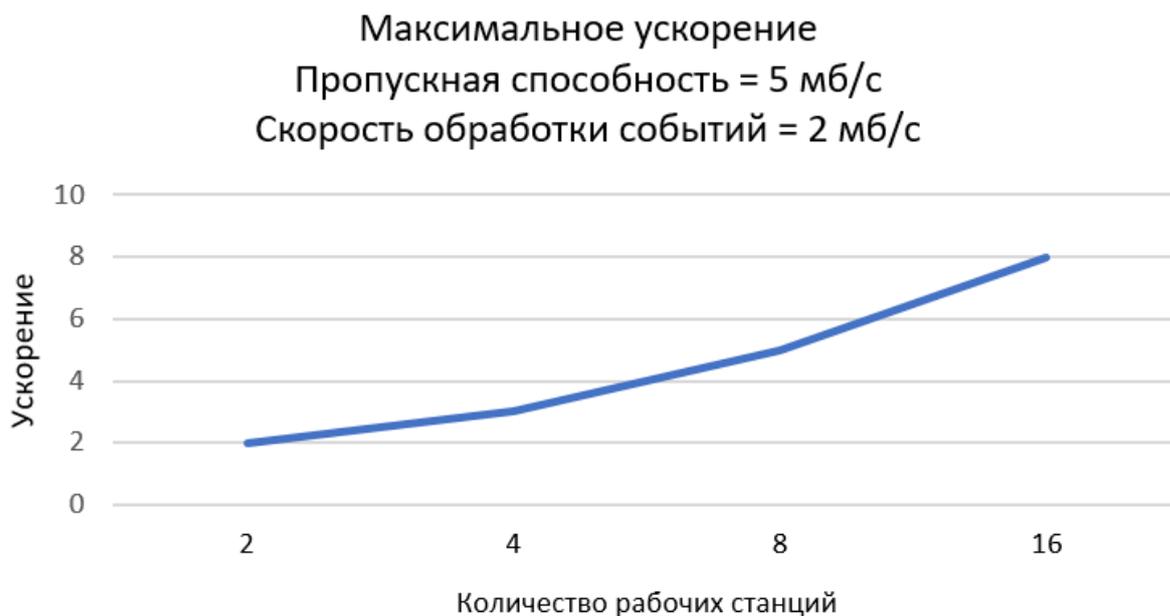


Рисунок 6 - Максимальное ускорение в сравнении с количеством рабочих станций

Как упоминалось ранее, рабочий возвращает время своего выполнения, в то время как мастер выполняет все найденные события до тех пор, пока входные файлы журнала событий не будут полностью проанализированы и все отправленные задачи не будут выполнены. Наблюдаемое ускорение было вычислено для теста путем деления суммы всех затраченных времен, возвращаемых каждым вызовом рабочей станции, на затраченное время выполнения мастером, умноженное на коэффициент нормализации, чтобы компенсировать разницу в скорости между машиной, на которой выполняется мастер, и узлами PlanetLab, на которых происходит выполнение теста.

Таким образом, основные экспериментальные результаты параллельной обработки файлов журналов приведены с точки зрения того, насколько близок каждый набор рабочих станций к достижению теоретически максимального ускорения (рисунок 6) для обрабатываемых задач различного размера и, таким образом, обеспечения наилучшего возможного времени обработки при распараллеливании обработки данных.

С этой целью на рисунке 7 показано графическое представление выдержки из этих результатов в относительном выражении для выборки конкретной задачи размером в 5 событий.



Рисунок 7 - Относительное ускорение в зависимости от количества работников для задачи размером в 5 событий

Как можно заметить, количество участвующих рабочих станций в работе прямо пропорционально влияет на эффективность работы системы в целом, поскольку достигается больший коэффициент продуктивности системы. Однако стоит учитывать, что прирост коэффициента продуктивности достигает своего максимума при определенном количестве рабочих станций. Для точного понимания количества рабочих станций, дабы достигнуть пика производительности и экономичности, рекомендуется отталкиваться от медианы максимального и среднего количества событий, где на одно событие, приходится в среднем 2-3 машины.

4.2 Результаты распараллеливания обработки файлов журнала с использованием JXTA

В этом подразделе представлены экспериментальные результаты, полученные после запуска grid-платформы на базе JXTA в узлах PlanetLab,

составленной из тестовых лог-файлов, демонстрирующих достигнутое ускорение.

Данный тест использует как большие объемы лог-информации (например, файлы ежедневных журналов), так и хорошо стратифицированные короткие выборки, состоящие из репрезентативных ежедневных периодов с различной степенью активности. Кроме того, в других тестах использовалось несколько файлов журнала с выбранным размером файла, образующих выборку каждой репрезентативной стратегии. Это позволило получить надежные статистические результаты с использованием простого в использовании размера входных данных.

Тест был сначала выполнен несколько раз на однопроцессорных машинах с обычными конфигурациями и различной рабочей нагрузкой, чтобы получить более надежные результаты в статистических терминах, включая размер файла, количество обработанных записей журнала и время выполнения наряду с другой базовой статистикой. Затем тот же тест был обработан параллельным образом JXTA, с использованием разного количества узлов, в частности, 2, 4, 8 и 16 узлов, используя узлы PlanetLab.

Параллельная эффективность, которая измеряет степень использования вычислительных ресурсов, задействованных в параллельных вычислениях, определяется как (2).

$$E(s, p) = S(s)/p, \quad (2)$$

где $E(s, p)$ – параллельная эффективность;

$S(s)$ – параллельное ускорение;

p – количество вычислительных ресурсов.

На рисунке 8 показаны графики зависимости времени обработки файлов журнала с количеством обрабатывающих процессоров.

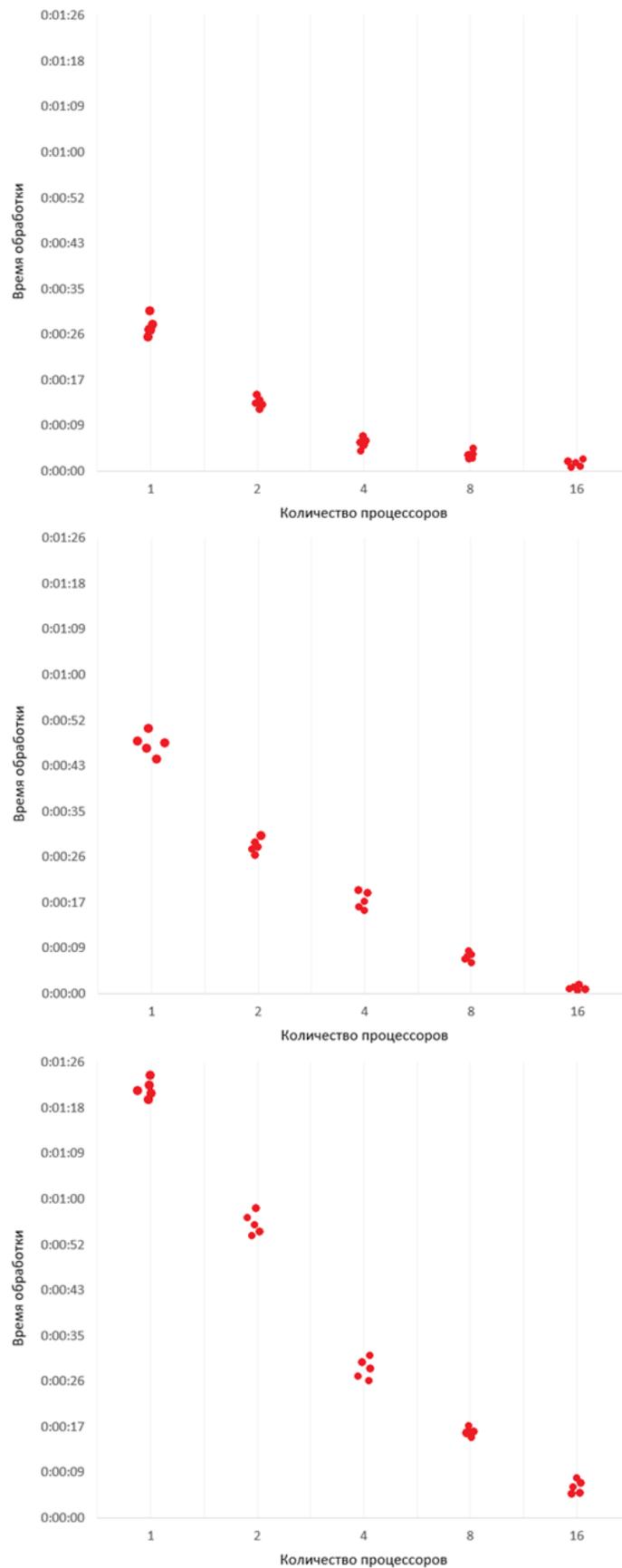


Рисунок 8 - Три результата времени выполнения для файлов журнала размером 12 МБ, 24 МБ и 36 МБ соответственно

Исходя из времени выполнения, представленного на рисунке 8, и приведенных ранее формул, полученные результаты показывают выигрыш в плане ускорения и эффективности.

4.3 Анализ результатов распараллеливания обработки файлов журнала

Анализируя экспериментальные результаты, полученные на прототипах grid, было обнаружено, что, с одной стороны, при определенных значениях размера задачи наблюдаемое ускорение было очень близко к теоретически достижимому максимуму. Следовательно, только при очень небольшом значении размера задачи влияние на ускорение может быть значительным из-за затрат на передачу данных. Однако также было замечено, что чем больше работников использовалось в тестах, тем ближе к теоретическому максимуму было ускорение, достигаемое небольшими задачами, и это увеличивалось до такой степени, что при достаточном количестве работников даже самые маленькие задачи (т.е. задача с одним событием) решение позволяло достигать значительного ускорения.

С другой стороны, однородное поведение, наблюдаемое в узлах PlanetLab, оправдывало решение о тестировании с одинаковым размером задачи для всех работников. Однако в реальной среде grid, размеры задач должны быть скорректированы в зависимости от случая с рабочим узлом, чтобы соответствовать динамически изменяющимся рабочим нагрузкам, с которыми могут экспериментировать узлы, и учитывать различные скорости работы машины. Однако было отмечено, что, хотя результаты этого эксперимента являются многообещающими, необходим более глубокий и точный анализ как первичного взаимодействия, происходящего между участниками в виртуальных классах, так и реальной совместной учебной деятельности на основе сложных параметров сотрудничества, таких как выполнение вышеупомянутых задач, функционирование группы и создание

«каркасов», как ожидается, будут генерировать гораздо большее количество и более сложных событий, чем те, которые использовались для экспериментов. Этот сценарий позволит в гораздо большей степени воспользоваться преимуществами, предоставляемыми grid-средой, и предоставит более полезные знания о фактической производительности, происходящей в процессе онлайн-обучения, а также поможет более удобно отслеживать и поддерживать участников обучения.

После этого исследования важно иметь в виду, какова мотивация всего этого исследования. С этой целью в этом последнем разделе обобщаются основные достигнутые результаты и очень кратко указывается, как их применять в контексте обучения.

До сих пор обсуждалось, как предоставление непрерывных знаний онлайн-командам в среде CSCCL может значительно улучшить групповую деятельность с точки зрения принятия решений, групповой организации, социальной вовлеченности, поддержки, мониторинга и так далее. В результате необходимо эффективно обрабатывать и анализировать большие объемы лог-информации, сгенерированной в результате совместного взаимодействия. Более того, чтобы знания, извлеченные из анализа, были полезны для информирования и обратной связи, пользователям должна предоставляться как разовая информация, быстро доставляемая в режиме реального времени, так и сложная, исчерпывающая, но структурированная отложенная информация, что еще больше повышает требования к обработке, выходящие за рамки требований.

Кроме того, веб-приложения, поддерживающие дистанционное обучение в режиме онлайн, также могут извлечь значительную выгоду из извлечения выбранных знаний из файлов журнала, для целей пользовательского моделирования. Действительно, эти приложения, благодаря высокой степени взаимодействия с пользователем, используют большие преимущества основанных на отслеживании методов пользовательского моделирования, таких как предоставление более широкой

и качественной поддержки пользователям веб-образовательных систем. Анализ информации, полученной в результате действий, выполняемых учащимися, является основной функцией моделирования поведения учащегося в процессе обучения, а также самого процесса обучения. Кроме того, построение моделей учащихся может помочь определить шаблоны навигации и адаптировать удобство использования системы к реальным потребностям учащихся, что значительно стимулирует процесс обучения.

Grid-подход, описанный в этом разделе, представляют собой важный шаг к преодолению жестких требований за счет уменьшения времени обработки большого объема файлов журналов, хранящих сложную информацию о событиях групповой активности. Полученные результаты и опыт, накопленный в ходе этих исследований, позволяют сделать вывод о том, что вопрос о том, выгоден grid или нет, будет в значительной степени зависеть от объема и структуры обрабатываемой информации. Таким образом, эти результаты побуждают продолжать работу над разработкой реальной рабочей реализации grid для решения проблемы обработки файлов журнала событий групповой активности. Идеи и опыт, собранные к настоящему времени, обеспечат необходимую основу и ресурсы для решения предстоящих задач в рамках этой диссертации.

4.4 Продвинутое развитие разработки программного обеспечения с использованием Библиотеки совместного обучения

Слияние парадигм и методов разработки программного обеспечения, описанных в 3 разделе (т.е. сервис-ориентированной архитектуры (SOA), разработки на основе моделей (MDD), универсального программирования (УП) и архитектуры продуктовых линеек (PLA)), привело к разработке модели БСО, которая предоставит разработчикам методы разработки программного обеспечения для эффективной и своевременной разработки приложений CSCL.

БСО основан на парадигмах сервис-ориентированной архитектуры (SOA) и универсального программирования (УП), чтобы обеспечить полное и эффективное повторное использование его общих компонентов в качестве основы для построения любого приложения для совместного обучения посредством реализации и концептуализации фундаментальных потребностей, существующих в любом приложении CSCL. БСО также обеспечивает полную поддержку распространения, возможности повторного использования, гибкости и интероперабельности как ключевых аспектов для удовлетворения текущих нефункциональных потребностей в разработке программного обеспечения в целом и, в частности, в области CSCL.

Существует много общего между повсеместными и сложными потребностями в совместном обучении и преимуществами, предоставляемыми SOA. В результате такого сопоставления SOA представляется наилучшим выбором для поддержки разработки БСО. Действительно, SOA расширяет возможности образовательных организаций за счет повышения гибкости их педагогических стратегий, которые могут постоянно адаптироваться, корректироваться и персонализироваться для каждой конкретной целевой учебной группы. Более того, SOA облегчает повторное использование успешного опыта совместного обучения и позволяет участникам совместного обучения легко адаптировать и интегрировать свои текущие передовые практики и существующие хорошо известные инструменты обучения в новые учебные цели. За последние годы CSCL превратился в сложную и обширную область.

Таким образом, применение принципов УП представляется хорошим выбором для разработки БСО путем определения тех частей, которые являются общими для большинства приложений CSCL.

Чтобы превратить БСО в эффективную программную платформу, ее разработка была основана на подходе Model-Driven Development (MDD). Эта парадигма очень хорошо сочетается с принципами УП и SOA, благодаря четкому разделению универсальной, многообразной, независимой от

технологии модели от различных, гибких, зависящих от технологии моделей реализации. Более того, веб-сервисы являются технологией внедрения, выбранной для БСО, учитывая широко принятые протоколы и стандарты, что представляет собой само обоснование этой технологии. Все эти стандарты представляют собой подходящий контекст для обеспечения интероперабельности и масштабируемости за счет использования больших преимуществ распределенных технологий.

Наконец, чтобы максимально автоматизировать переход от платформенно-независимой модели (PIM) к соответствующей платформенно-специфичной модели (PSM), последние результаты исследований в этой диссертации приводят к работе с файлами XMI, которые представляют собой файлы с XML-тегами в результате кодирования UML-диаграмм. В сочетании с таблицами стилей XSL можно превратить файлы XMI PIM в файлы WSDL, которые представляют собой входные данные для рабочей среды веб-сервиса, чтобы преобразовать их в архитектурный проект на определенном языке (PSM).

Несоответствие стандарту существующих инструментов UML, является основной проблемой, с которой придется столкнуться в будущем, а также с тем, как обеспечить более полную и детальную реализацию желаемого PSM. Эти открытые вопросы должны быть исследованы в качестве дальнейшей работы в контексте данной диссертации.

Заключение

Одной из ключевых проблем при разработке приложений CSCL является анализ данных взаимодействия, который является основной функцией для поддержки обучения и оценки в средах CSCL. Он опирается на информацию, полученную в результате действий, выполняемых участниками в ходе процесса совместной работы. Эффективное внедрение этой информации и извлеченных знаний в приложения CSCL создает основу для повышения осведомленности о мониторинге и обратной связи для достижения успешного процесса обучения в условиях совместной работы.

Современные среды CSCL больше не зависят от однородных групп, статичного контента и ресурсов, а также от единой педагогики, но в этом контексте необходима высокая индивидуализация и гибкость, а это означает, что методы совместного обучения необходимо постоянно адаптировать, корректировать и персонифицировать для каждой конкретной целевой учебной группы. Удовлетворение этих потребностей в области CSCL представляет собой серьезную проблему для исследовательского сообщества.

Магистерская диссертация посвящена моделированию структуры эффективного управления и анализа данных взаимодействия для создания универсального механизма разработки программного обеспечения на основе повторного использования компонентов приложений CSCL.

Выполненные в работе научные исследования представлены следующими основными результатами:

- проанализирован процесс сбора, обработки, анализа и представления информации и знаний, полученных в результате групповой деятельности, в приложениях CSCL. Как показал анализ, главным недостатком известных CSCL решений является их узкая направленность, главная задача которых, решить конкретную, возникшую «у себя», проблему. На основании проведенного анализа, в качестве решения проблемы была разработана структура эффективного

управления и анализа данных взаимодействия, которые генерируются в ходе практики CSCL;

- усовершенствован метод разработки программного обеспечения на основе повторного использования компонентов приложений CSCL для полуавтоматического генерирования реализаций на основе независимых спецификаций, что совместно с разработанной моделью вычислительной системы в форме общей спецификации, позволяет добиться эффективного повторного использования системы за счет сниженной стоимости повторной разработки, дистанцирования от конкретных платформ и языков программирования, а также, за счет сниженных затрат времени на развертку системы;
- проведена экспериментальная апробация усовершенствованного механизма управления платформой сообщества для поддержки интегрированного проектирования обучения на основе открытых источников, которая показала эффективность использования grid-архитектуры в качестве вычислительной структуры.

Таким образом, в работе решена научно-практическая проблема разработки механизмов управления платформой сообщества преподавателей для поддержки интегрированного проектирования обучения.

Гипотеза исследования подтверждена.

Проведенная работа имеет большую значимость, так как благодаря использованию вышеописанных решений возможно повышение универсальности и эффективности систем CSCL.

Список используемой литературы

1. Итинсон К. С. Облачные технологии в образовании: концепция и реальность. Балтийский гуманитарный журнал. – Курск, 2020. – 90-92 с.
2. Коневщинская О. Э. Информационные технологии и средства обучения. – 2017. – 6 с.
3. Латышева Л. П., Скорнякова А. Ю., Черемных Е. Л. О применении интерактивных методов в дистанционном обучении математике студентов педагогического вуза. Гуманитарные исследования. педагогика и психология. – Пермь, 2020. – 26-35 с.
4. Мадрацо Л. Пересечение взаимодействий между архитектурой и средами. Педагогическая модель для современного образования. Архитектура и современные информационные технологии. – Москва, 2018. – 2 с.
5. Хасанова Г.Ф., Шагеева Ф.Т., Иванов В.Г. Групповая проектная деятельность студентов в процессе изучения курса «основы инженерной психологии» в исследовательском университете. Сетевое взаимодействие как эффективная технология подготовки кадров. – Казань, 2015. – 159-163 с.
6. Anido-Rifon L., Llamas M., Fernandez M. J., Caeiro M., Santos J., Rodriguez J. A Component Model for Standardized Web-based Education. Conference Proceedings of 10th International World Wide Web Conference. – 2015. - 86-95 сс.
7. Alario-Hoyos C., Bote-Lorenzo M. L., Gómez-Sánchez E., AsensioPérez J. I., Vega-Gorgojo G., & Ruiz-Calleja A. GLUE!: An architecture for the integration of external tools in virtual learning environments. Computers & Education. - 2017. - 122–137 сс.
8. Bacelo Blois A., Becker K. A Component-Based Architecture to Support Collaborative Application Design. Groupware: Design, Implementation and Use. - 2017. - 134-143 сс.
9. Bote Lorenzo M.L., Gomez Sanchez E., Vega Gorgojo G., Dimitriadis Y., Asensio Perez J.I., Jorriñ Abellan I.M. Gridcole: a tailorable grid service-based

system that supports scripted collaborative learning. *Computers and Education*. - 2021.

10. Bratitsis T. and Dimitracopoulou A. Monitoring and Analysing Group Interactions in asynchronous discussions with the DIAS system. *Computer Science*. – 2020. - 54-61 cc.

11. Calvo A., Turani A. E-Learning frameworks. *E-Learning, design patterns and pattern languages*. Sense Publishers. - 2017. - 49-64 cc.

12. Daradoumis T., Xhafa F., and Juan Perez A. A Framework for Assessing Self Peer and Group Performance in e-Learning. *Self, Peer, and Group Assessment in e-Learning*. - 2019.

13. De Smet M., Van Keera H. and Valcke M. Blending asynchronous discussion groups and peer tutoring in higher education: An exploratory study of online peer tutoring behaviour. *Computers and Education*. - 2022. - 207-223 cc.

14. Dimitriadis Y. A., Asensio J. I., Toquero J., Estebanez L., Martin T.A., and Martinez A. Towards a component-based system for the CSCL domain. *Proceedings of the Symposium on Informatics and Telecommunications*. -2016

15. Foster I., Kesselman C., Tuecke S. *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*. *International Journal of Supercomputer Applications and High-Performance Computing*. – 2015.

16. Garcia I., Gros B., & Noguera I. The relationship between current technological affordances and the design of learning activities for collaborative knowledge construction. *Cultura Y Educación*. - 2016. - 395–417 cc.

17. Goodyear P., & Retalis S. *Technology-Enhanced Learning, design patterns and pattern languages*. The Netherlands: Sense Publishers. - 2015.

18. Gutl C. "Moving Towards a Generic, Service-based Architecture for Flexible Teaching and Learning Activities. *Architecture Solutions for E-Learning Systems*. – 2021. - 1-24 cc.

19. Hernández-Leo D., Asensio-Pérez J. I., Dimitriadis Y., & Villasclaras-Fernández E. Pattern languages for generating CSCL scripts: from a conceptual model to the design of a real situation. *E-learning, design patterns and pattern*

languages. - 2019. 49–64 cc.

20. Hernández-Leo D., Chacón-Pérez J., Prieto L. P., Asensio-Pérez J. I., & Derntl M. Towards an integrated learning design environment. The 10th European Conference on Technology Enhanced Learning. - 2015. - 448–453 cc.

21. Hew K.F. and Cheung W.S. Attracting student participation in asynchronous online discussions: A case study of peer facilitation. Computers and Education. – 2008.

22. Laforcade P., Choquet C., Nodenot T., Pierre-Andre, C. (2021). Model-Driven Engineering (MDE) and Model-Driven Architecture (MDA) applied to the Modeling and Deployment of Technology Enhanced Learning (TEL), Systems: promises, challenges and issues. Architecture Solutions for E-Learning Systems. – 2021.

23. Marques J. M., Navarro L. LaCOLLA: A Middleware to Support Self-sufficient Collaborative Groups. Computers and Artificial Intelligence. – 2020.

24. Martinez A., de la Fuente P., and Dimitriadis Y. Towards an XML-based representation of collaborative interaction. International Conference on Computer Support for Collaborative Learning. – 2017. - 379-384 cc.

25. Mor Y. SNaP! Re-using, sharing and communicating designs and design knowledge using scenarios, narratives and patterns. Handbook of Design in Educational Technology. UK. London. - 2018.

26. Nielsen L. Personas. The Encyclopedia of Human-Computer Interaction, 2nd Ed. Denmark. Aarhus. - 2018.

27. Pena-Shaff J. and Nicholls C. Analyzing student interactions and meaning construction in computer bulletin board discussions. Computers and Education. - 2018. - 243-265 cc.

28. Schellens T. and Valcke M. Fostering knowledge construction in university students through asynchronous discussion groups. Computers and Education. - 2020. - 349-370 cc.

29. Schrire S. Knowledge building in asynchronous discussion groups: Going beyond quantitative analysis. Computers and Education. - 2020. - 49-70 cc.

30. Vdovjak R., & Houben G. RDF-based architecture for semantic integration of heterogeneous information sources. In The International Workshop on information integration on the Web. The Netherlands. Eindhoven. - 2015. - 51-57 cc.

31. Villasclaras-Fernández E. Incorporating assessment in a pattern-based design process for CSCL scripts. Computers in Human Behavior. - 2018. - 1028–1039 cc.

32. Zumbach, J., Schonemann, J., and Reimann, P. Analyzing and Supporting Collaboration in Cooperative Computer-Mediated Communication. Computer Supported Collaborative Learning 2005: The Next 10 Years! – 2019. 758-767 cc.